

2 - AGOSTO 2003

panorama

LOS USUARIOS DE SPECTRUM

zona www

EL MUNDO DEL SPECTRUM 2.0

emulación

INTRODUCCIÓN A LA EMULACIÓN

análisis

KUNG FU MASTER

PEKING

MIGHTY MAGUS

HISTORIA ILUSTRADA DEL SPECTRUM

y mucho más...

www.speccy.org/magazinezx





Redacción

Santiago Romero
Miguel G. Prada
Federico Álvarez
Pedro J. De Celis – Historia Ilustrada del Spectrum

Ilustración de portada

Juanje Gómez

Colaboraciones en este número

Pablo Suau
Víctor Salado

Contacto

magazine@speccy.org



Aquí estamos en esta segunda entrega de la revista Magazine ZX dispuestos a amenizar el verano con refrescantes artículos dedicados al mundo del Spectrum.

En nuestra anterior y primera entrega os presentamos una declaración de intenciones de cómo pretendíamos que fuera esta publicación y en qué se originaba. Afortunadamente fue bien acogida y es por ello que este mes volvemos a la carga con nuevos contenidos, artículos más completos y ganas de continuar con la elaboración de la nueva revista. Nos han llovido mensajes de muchos lectores con ideas, sugerencias, peticiones de colaboración y, sobre todo, agradecimiento por llenar un hueco que existe desde el cierre de Microhobby. Personalmente en la redacción de Magazine ZX pensamos que es muy difícil llegar a ser una revista de tanta calidad como MH, pero no se nos escapa que es algo que se puede conseguir mes a mes, a lo largo del tiempo. Para empezar se ha obtenido el permiso directo de Martijn van der Heide (de World Of Spectrum) y de Juan Pablo López Grao (de Spanish Spectrum Archive) para enlazar directamente a los juegos desde nuestros artículos.

La orientación del número de este mes está principalmente centrada en la descripción para los noveles de los secretos básicos de la Emulación, así como en el análisis del controvertido Kung Fu Master de US Gold. Tendremos la oportunidad de estudiar un par de visiones diferentes del mismo juego, en un duelo de opiniones a favor y en contra.

Por otra parte, la sección de Hardware de este mes finaliza el montaje de la disquetera de 3"1/2 para +3, para aprovechar en su totalidad todo el espacio que nos ofrecen los discos de baja densidad de PC, hasta 720KB de almacenamiento.

Finalmente, este mes os ofrecemos la continuación del curso de BASIC junto al comentario habitual de los juegos semi-desconocidos de rigor, sin olvidarnos de "La Historia Ilustrada del Spectrum", de la mano de nuestro amigo Pedrete.

Como siempre, aceptaremos todo tipo de críticas, colaboraciones o ideas, que deberéis enviarnos a **magazine@speccy.org**. Cualquier comentario será bienvenido, ponte en contacto con nosotros y de esta forma conseguiremos una revista más viva, dinámica y útil para todos.

Este Agosto, entre piscinazo y piscinazo no puede faltar la lectura de este nuevo número de Magazine ZX. Esperamos que lo disfrutéis.

Redacción de MAGAZINE ZX



Redacción de MAGAZINE ZX

¿A dónde va el mundillo del Spectrum? ¿Quién tira de él? ¿Cuales deberían ser nuestros objetivos como usuarios de Spectrum para preservar su legado y continuar ofreciendo cosas nuevas?

En Junio de este año (2003) hubo un hilo en el grupo de news **es.comp.sistemas. sinclair** titulado "Desorganización?" que venía a preguntar al grupo si los usuarios de Spectrum estábamos organizados con respecto a preservar y potenciar los modelos de microordenadores de Sinclair. De ese hilo se derivan 2 discusiones de vital importancia, que son:

1. ¿Estamos los usuarios de Spectrum organizados? ¿Cómo son esos usuarios?
2. ¿Cómo debemos actuar los usuarios para preservar lo que nos ha aportado el Spectrum, y crear nuevos contenidos que lo continúen impulsando?

En este artículo trataremos el primer punto.

En otros sistemas como MSX y Amiga existe un verdadero movimiento que, pese a algunas guerras internas, promociona quedadas, "salones" donde se presentan nuevos videojuegos, acude a "ferias" de Informática con sus propios Stands, y, en general, se organiza de una manera correcta para aprovechar los recursos del colectivo en favor de potenciar los sistemas que "defienden" y utilizan. En el caso del Spectrum a nuestro modo de ver hay distintos tipos de usuarios. Es complicado generalizar, pero podemos dividirlos en:

- Usuarios muy activos:

  Son aquellos que realizan una labor intensa en pro del Spectrum, como programar nuevos juegos, crear webs relacionadas, abarcar proyectos de preservado (escaneos, transcripciones a HTML/PDF, conversiones a TZX/TAP o a listados de texto). Curiosamente, en general son los que menos se pronuncian en el IRC o las NEWS, donde sólo aparecen para hablar de cuestiones realmente importantes para el mundillo, o para anunciar actualizaciones (aunque alguno de ellos puede ser activo en estos medios).

- Usuarios activos del sistema:

  Son aquellos que siguen usando el sistema, realizan de vez en cuando tareas a favor del Spectrum (como enviar algún artículo, pasar algún fichero a disco o tzx, o colaborar en alguna web o proyecto). Al no dedicar todo su tiempo libre a proyectos como en el caso de los usuarios del grupo anterior, suelen aparecer más en los foros de discusión. Estos usuarios son también importantes (como los anteriores), ya que son aquellos que aprovechan realmente el trabajo realizado por los primeros y lo revierten en los foros e IRC.

- Usuarios:

  Son aquellos que utilizan el sistema, principalmente a través de emuladores. No realizan aportaciones esenciales a las webs y proyectos, pero los aprovechan. Son los "destinatarios" de todo lo realizado por los 2 grupos de usuarios que he nombrado hasta el momento.

- Esponjas:

  Son aquellos usuarios dispuestos a recoger todo lo que puedan sin aprovecharlo ni dar nada a cambio. No usan el sistema ni en máquina real ni en emulador, y simplemente se dedican a acaparar todo lo posible en forma de CDs de juegos, revistas, escaneos o máquinas. Apenas hablan en las news ni el IRC. No tienen Web ni aportan nada.

- Ausentes:

  Estos usuarios son todos los restantes y que no entran en ninguno de los grupos anteriores sencillamente porque no conocen la existencia de las news, del IRC o de las diferentes páginas sobre el Sistema. Tienen un Spectrum o lo tuvieron, tienen interés y material, pero no han llegado a contactar con gente de los anteriores grupos para terminar de desarrollar ese interés y convertirse en usuarios proactivos del sistema.

¿Para qué sirve esta clasificación? Simplemente es un reflejo (nuestra visión particular) de cualquier grupo de personas dedicada a una afición común.
¿Quién tira del mundillo del Spectrum? Obviamente los artífices más importantes son los del grupo 1. Son los escaneadores, los "conversores de programas" (en pro de preservar las cintas), los que hacen Webs usando su tiempo libre para proporcionar a otros información, etc. Los siguen los del grupo 2 y los del 3, que son quienes dan vida a los foros de las webs, IRC y news. Lamentablemente la gente del primer grupo es muy escasa, y por si fuera poco en ocasiones se va desligando del mundillo ya que su dedicación a veces no es reconocida y su trabajo acaba sencillamente en los discos duros de los usuarios esponja. Cuando piden ayuda para llevar a cabo un determinado proyecto, son pocos los que acaban ofreciendo ayuda real, y finalmente éstos tienen que realizar todo el trabajo por su parte, como ocurrió con el proyecto de escaneo de las revistas Microhobby: una sóla persona ha realizado el escaneo de los miles de páginas de las 217 Microhobbies más especiales, por su propia mano, sin ayuda, y en un tiempo récord. Si la gente del grupo 1 fuera incombustible tendríamos movimiento en el mundillo para rato, pero llega un momento en que esta gente se cansa y aparece cada vez menos. Es necesario un relevo generacional (normalmente con gente del segundo grupo) para que no se muera el mundillo o se estanque. Y ese relevo requiere implicación y dedicación, algo que no es muy compatible con la mayoría de nosotros, que disponemos de un tiempo escaso.

Lamentablemente es muy fácil encontrar gente del grupo "esponja", pero la ausencia más triste es la existencia del último grupo, el de los "usuarios perdidos" que no conocen los medios con los que se podrían relacionar con otros compañeros con las mismas aficiones. Una demostración de la gran presencia de este tipo de usuarios se produjo cuando USA-SOFT (un usuario de Spectrum) realizó una recopilación de juegos de Spectrum en formato Z80, autoarrancable desde CD, y que utilizaba el Spectrum Games Database como front-end y el emulador X128 para la emulación. Dejando de lado la calidad de la recopilación (X128 no era el emulador más aconsejable para equipos Windows, y .Z80 no es el mejor formato disponible para preservar los juegos, en mi opinión), USA-SOFT puso la recopilación de juegos en un conocido sistema de compartición de ficheros P2P (edonkey) como un único fichero ZIP que al descomprimirse podía grabarse en CD y disfrutar de 10.100 juegos de Spectrum desde el mismo CD.

Desde la publicación de ese CD no cesaron de aparecer nuevos visitantes en el canal #spectrum del IRC-Hispano preguntando sobre el CD, cómo hacer

que funcionara en XP (X128 no funciona en esa versión del S.O. de MS), cómo cargar las cintas en formato TAP del CD, etc. Estos usuarios no conocían ni las news, ni las diferentes Webs de Spectrum. No conocían World Of Spectrum ni ninguna otra web relacionada. Su única vinculación con el Spectrum había sido encontrar ese fichero ZIP en edonkey, que en el mismo nombre de fichero instaba a entrar en el canal #spectrum. Es decir, estos usuarios poseían interés en el Spectrum, y habían buscado la palabra Spectrum en edonkey, encontrando el fichero ZIP de USA-SOFT. Esa masa de usuarios desconoce la presencia del Spectrum en los diferentes medios, y había llegado a ellos por un medio que ninguno de nosotros había explotado hasta el momento. Y todavía no he nombrado a aquellos que tuvieron Spectrums y no son habituales de Internet (es decir, que no se conectan de forma habitual desde sus casas).

Una de las principales misiones de los Spectrumeros de hoy debería ser el encontrar y orientar a estos usuarios y mostrarles las diferentes webs, proyectos y puntos de encuentro disponibles. Estos nuevos usuarios entrarán en diferentes grupos de los anteriormente descritos. Está claro que cantidad no equivale a calidad, y muchos de estos usuarios serán "esponjas" cuya única aportación será descargar e incluso distribuir sin permiso de los autores (vía edonkey) los contenidos que cuelgan de las diferentes Webs.

Pero también podemos encontrarnos sorpresas como por ejemplo gente cercana a nuestras localidades y por lo tanto la gestación de las primeras quedadas locales, o gente dispuesta a escanear sus libros o pasar su cintas, o simplemente gente dispuesta a "estar" y "participar".

Tenemos una gran diversidad de Webs y servidores de IRC, pero el punto de encuentro común y principal de todos nosotros debería ser las news, y en concreto el grupo **es.comp.sistemas.sinclair**. Un grupo de news está mucho más allá que los intereses de sus usuarios, o de las webs de los mismos. Está más allá de las redes de IRC (privadas o públicas) y es relativamente fácil de encontrar. El grupo de news existe desde hace muchos años, y seguirá estando ahí aún cuando haya gente que se vaya del mundillo. En mi opinión lo más aconsejable para mejorar el mundillo del Spectrum sería publicitar las news de la siguiente forma:

- Mostrar el acceso y uso de las news a los usuarios.
- Dar publicidad del grupo **es.comp.sistemas.sinclair** en todos los sitios (recopilaciones de CDs a la venta o en edonkey, o regalados a los amigos, páginas web, servidores de IRC, mensajes en otros grupos de news, firmas de correo).
- Acoger bien a la gente recién llegada, siempre respetando las *reglas de nettiqueta* básicas en las news. Mostrar estas reglas a los nuevos para una correcta convivencia.
- Desde este punto de encuentro común, mostrar todos los recursos y proyectos disponibles.
- Desde ese mismo momento, las personas con proximidad geográfica que así lo deseen ya tendrán un medio de comunicación que los ha dado a conocer entre ellos, aparecerán quedadas, y tal vez algún día alguna Asociación de Usuarios.

Ésta ha sido una crítica algo larga y en algunos puntos bastante ácida y sin censura, pero necesaria para concienciarnos de que sin implicación, este mundillo puede morirse. Está claro que el Spectrum es un ordenador sin crecimiento comercial, pero su muerte o su expansión dependerá exclusivamente de nuestra actitud y esfuerzo.



SANTIAGO ROMERO, PABLO SUAU Y VÍCTOR SALADO

Este mes presentamos un doble análisis de **Kung Fu Master**, por parte de Pablo Suau y Santiago Romero, frente a frente en un particular duelo dialéctico. Además, de postre, un par de sabrosos análisis, el primero del desconocido y adictivo **Peking** y el segundo del también poco conocido **Migthy Magus**, a cargo de Víctor Salado.

KUNG FU MASTER



Trucos:

POKE 27982,0 : Vidas infinitas.
POKE 37400,0 : Tiempo infinito.
POKE 43912,58 : Energía infinita.

Otros Comentarios:

Crash nº 31, página 16:
ftp://ftp.worldofspectrum.org/pub/sinclair/magazines/Crash/Issue31/Pages/Crash3100016.jpg

Crash nº45, páginas 38-39:
ftp://ftp.worldofspectrum.org/pub/sinclair/magazines/Crash/Issue45/Pages/Crash4500038.jpg
ftp://ftp.worldofspectrum.org/pub/sinclair/magazines/Crash/Issue45/Pages/Crash4500039.jpg

Your Sinclair: Guide to Beat-em-ups:
http://www.ysrnry.co.uk/articles/completeguidetobeatemups.htm

Comentario de **Pablo Suau**

Muchos de nosotros, cuando vimos la máquina de salón de Kung-Fu Master, quedamos enganchados y deseábamos (como en muchas otras ocasiones había ocurrido y ocurriría con otras máquinas) que alguien creara una conversión para nuestro Spectrum. Y así fue como llegó el día en que US Gold nos trajo el Kung-Fu Master a casa, pero... ¿habría valido la pena la espera? ¿Nos encontramos ante una buena conversión?



El juego recién cargado

Lo cierto es que la historia no tiene excesivo guión, y de hecho, peca de poca originalidad. En el juego nos introducimos en la piel de un maestro del Kung-Fu que debe infiltrarse en el templo de un malvado mago y, tras superar cinco pisos llenos de peligros y enemigos, rescatar a una damisela en apuros. Al final de cada piso nos espera un terrible enemigo a batir (bueno, no tan terrible).

Cuando cargamos el juego nos llevamos una serie de decepciones. Para el que no conozca la recreativa, se trata del típico *beat'em up* de vista lateral en el que avanzamos de forma horizontal por cada uno de los pisos, debiendo eliminar los enemigos que encontramos por el camino mediante el empleo de puñetazos y patadas, y esquivando una serie de obstáculos. Los maravillosos gráficos y sonidos de la máquina han desaparecido para dar lugar a unos sprites y un mapeado no tan maravillosos. Los movimientos son algo bruscos y no demasiado naturales. Y cuando nos acercamos a los enemigos vemos una mezcla de atributos que hace bastante complicado diferenciar nuestro personaje del malo en cuestión.

Yendo más allá del aspecto externo, y aunque no basándose una idea no demasiado novedosa y original, podemos decir que desde las primeras partidas estaremos inmersos en un juego más entretenido y adictivo de lo que parece en un principio. Aunque pueda costar un poco eliminar a nuestros primeros contrincantes, una vez que nos hacemos con los golpes siempre querremos avanzar más y más. La dificultad de los primeros niveles no es demasiado elevada, y los enemigos que nos esperan al final de cada piso no demasiado fieros, pero dicha dificultad va creciendo paulatinamente, haciendo que siempre nos queden ganas de intentar llegar un poco más lejos. El número de vidas es de 3, aunque podrá ir aumentando con la puntuación y cada vez que superemos un piso. Es un número más que adecuado.



No todos los enemigos son humanos

Con respecto a los enemigos, los podemos encontrar de varios tipos. Evidentemente no nos aparecerán todos desde un principio y, aunque no hay demasiada variedad, al menos evita que nos aburramos con el fondo, que siempre es el mismo. Los más habituales serán aquellos que se limitarán a avanzar hacia nosotros restándonos energía en el caso de que colisionemos con ellos, pero también podremos encontrar otros que lanzan cuchillos, enanos saltarines, o incluso dragones y serpientes. Uno de los enemigos más importantes es el tiempo, pues dispondremos de una cantidad limitada para superar cada nivel. Pero no debemos temer, pues si avanzamos con decisión y sin entretenernos demasiado podremos llegar al enemigo final del piso con bastante holgura.

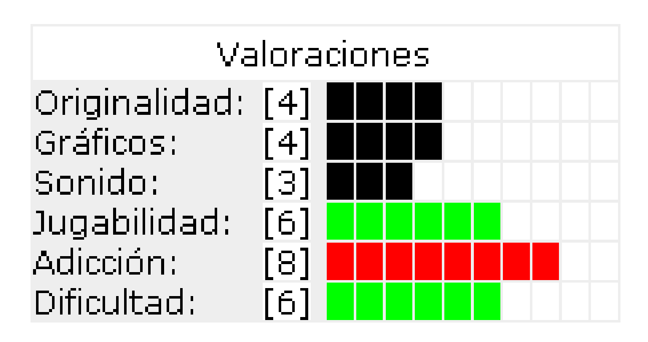


El 'terrible' enemigo que nos espera al final del segundo piso en acción

Los controles son configurables, permitiendo definir nuestras propias teclas o utilizar un joystick. Otro aspecto configurable es el nivel de inicio. Podemos escoger el piso desde el que queremos comenzar nuestra aventura, muy útil

si ya estamos cansados una y otra vez de superar los primeros niveles. Por último, pueden ser dos los jugadores que se enfrenten al malvado mago, pero por turnos.

En resumen: un juego infravalorado por su aspecto externo y sus deficiencias técnicas, pero que puede llegar a proporcionarnos bastante diversión y ser muy adictivo cuando le cojamos el tranquillo. La conversión no es brillante, pero es recomendable darle una oportunidad a pesar de la primera impresión que nos pueda causar.



Valoraciones	
Originalidad:	[4]
Gráficos:	[4]
Sonido:	[3]
Jugabilidad:	[6]
Adicción:	[8]
Dificultad:	[6]

Comentario de **Santiago Romero**

Kung-Fu Master es la conversión a Spectrum de una conocida y adictiva recreativa en la cual tenemos que rescatar a nuestra dama cautiva en un templo de 5 plantas defendido por fieros guerreros.



Pantalla de carga de KFM

Antes de comentar los aspectos técnicos del juego, veamos los puntos comunes y teóricos relativos al mismo.

Según la historia del juego, con nuestros conocimientos de Kung Fu debemos abrirnos paso por las 5 plantas del templo golpeando a todos los enemigos que aparezcan ante nosotros hasta rescatar a nuestra amada.

Para ello disponemos de golpes como la patada y puñetazo en posición normal, agachados, o el salto y patada (con este golpe podemos avanzar mientras atacamos, pero resulta bastante complicado acertar al enemigo, y especialmente inútil si se acercan varios simultáneamente).

El menú nos permite seleccionar entre el modo de 1 ó 2 jugadores (alternativos), cambiar la configuración de teclado y joystick, y comenzar el juego. Comenzamos la partida con 3 vidas, y podemos obtener nuevas vidas cada 40.000 puntos. Nuestra energía vital la mide una barra que disminuye con el contacto con los enemigos, y que nos hará perder una vida si se agota. También podemos perder una vida si el tiempo se acaba.



Nuestro héroe enfrentándose a sus enemigos

Tanto en la versión de Spectrum como en la recreativa, comenzamos en la parte derecha del templo y avanzamos hacia la izquierda (en el primer nivel, ya que al subir la primera escalera cambia el sentido de juego), deshaciéndonos de los enemigos que nos salen al paso. Estos enemigos se pueden eliminar fácilmente con un único golpe, pero si fallamos y llegan a tocarnos, nos quitarán energía hasta que nos zafemos de ellos. Existen diferentes tipos de enemigos, incluyendo una serie de guardianes que poseen cuchillos y no dudarán en arrojárnoslos si no los golpeamos a tiempo. Otros enemigos, como las

serpientes o dragones, plantearán más problemas. Aparte de ellos, en cada nivel existe un guardián que nos impide el paso a la escalera que da acceso al nivel superior. Ni qué decir tiene que los enemigos más poderosos son los que más puntos nos otorgarán al derribarlos. Existen 5 niveles de dificultad basados en la velocidad y número de enemigos, entre otros factores.

Hasta aquí la teoría. Pero pasemos a la práctica. Para empezar, nada más cargar el juego y después de ver la pantalla de presentación, se nos ofrece un menú con una melodía insoportable. Una vez redefinidas las teclas comienza el juego en el cual vemos como los bonitos sprites de la recreativa original se han convertido en manchas borrosas monocolor que se mueven sobre un fondo bastante colorido y con scroll no continuo (pantalla a pantalla). Por otro lado, durante gran parte de la ejecución del juego hay problemas de atributos y parpadeo de sprites (por ejemplo, al saltar y pasar sobre las ventanas).



Ese pegote de píxeles del centro de la pantalla es el protagonista junto a varios enemigos

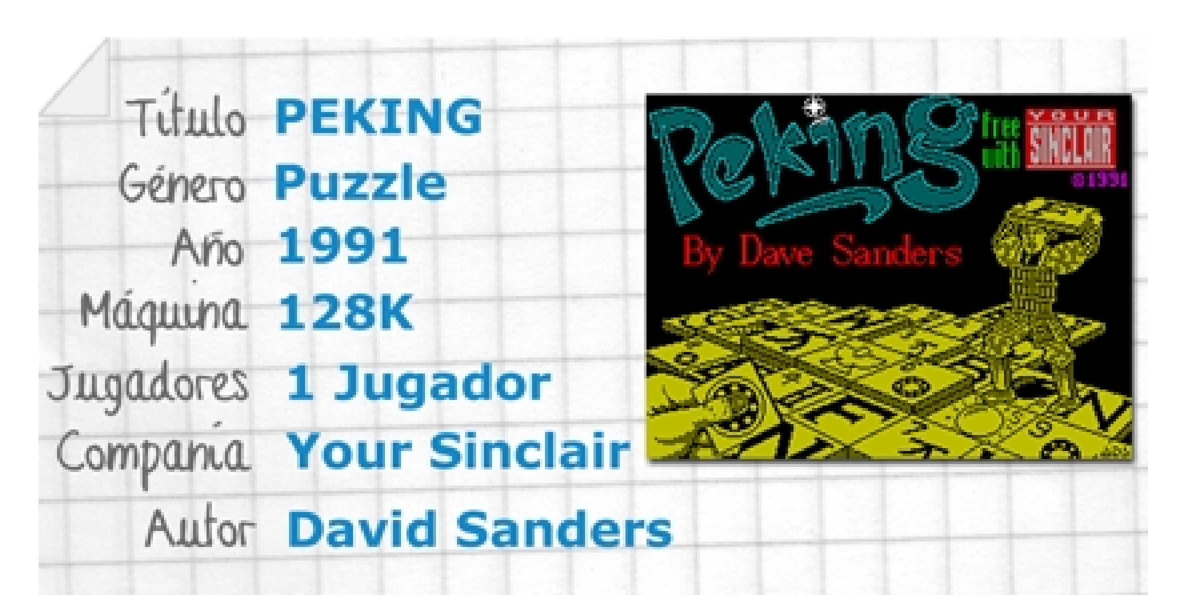
El teclado responde en general bastante bien a los movimientos, no así a los golpes, de modo que cuando se nos acercan enemigos, en ocasiones nuestra pulsación de golpeo es ignorada si coincide con el scroll de pantalla. La mayoría de los golpes fallarán su blanco y el enemigo nos alcanzará, haciéndonos perder energía (algo que no pasaba en la máquina original, donde los enemigos salían por centenares y eran bastante fáciles de batir). Las patadas voladoras son totalmente inútiles y sólo nos conducen a caer encima de todos los enemigos (tal vez matando al primero de ellos).

El contrapunto lo pone la música durante el juego, bastante decente en algunas melodías, aunque otras son absolutamente insoportables.

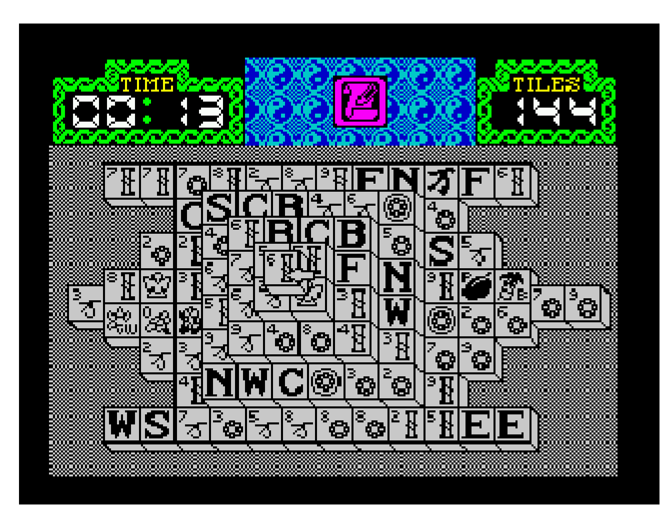
En resumen: una triste conversión de la maravillosa recreativa original. Es posible intentarlo durante unas partidas iniciales simplemente por la curiosidad de ver hasta dónde llegamos manejando al "montón de pulgas cabreadas" que parece el personaje, pero es muy difícil pasar de las primeras partidas con este juego.

Valoraciones	
Originalidad:	[3]
Gráficos:	[3]
Sonido:	[4]
Jugabilidad:	[3]
Adicción:	[3]
Dificultad:	[9]

PEKING



Peking es una producción amateur prácticamente desconocida en España que permite al usuario realizar una partida individual al clásico juego de tablero 'Mahjonng'. Su desconocimiento se basa en que no fue publicado por ningún sello como juego comercial, sino que apareció en la cara A del magazine Your Sinclair nº 73. Hemos indicado que el juego era amateur, pero bien podría haber sido publicado de forma comercial debido a la calidad y acabado del mismo.

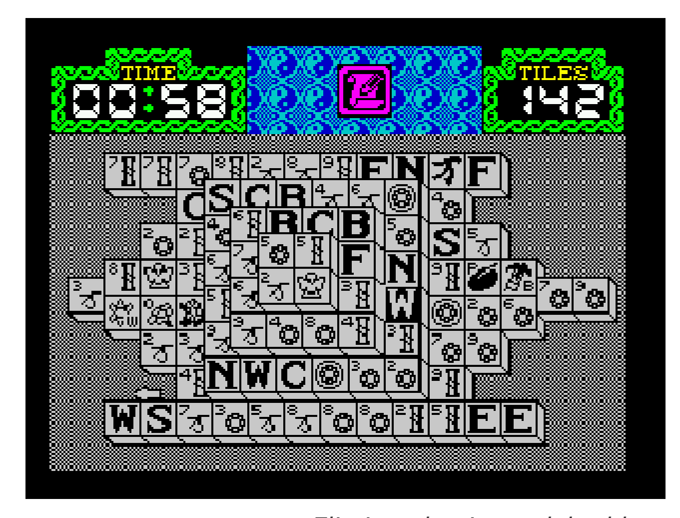


El tablero de juego del Mahjonng

El juego permite jugar a la versión individual del Mahjonng (que se puede jugar también con 4 jugadores en una modalidad de juego parecida a lo que es nuestro dominó, pero más complicada). En la versión de 1 sólo jugador (que es la que aparece con más frecuencia en la mayoría de ordenadores, a modo de solitario), se esparcen todas las piezas de mahjonng formando una figura, todas ellas con las caras de las piezas hacia arriba. Cada una de estas 144 piezas tiene un dibujo determinado. Están organizadas en varios palos, del 1 al 9. Además de estas fichas están las de las estaciones, los puntos cardinales, etc, llegando así al total de 144. El objetivo del juego es limpiar el tablero en su totalidad, eliminando pares de piezas en cada jugada.

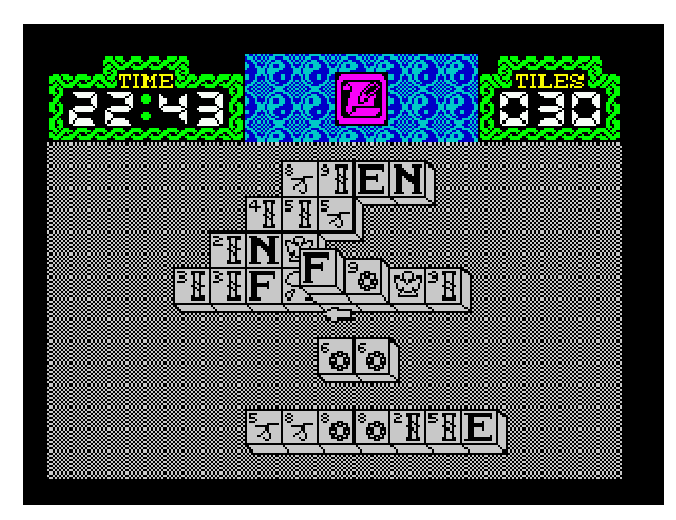
¿Qué piezas podemos eliminar? Sencillo: se eliminan piezas iguales (por ejemplo 2 piezas marcadas con el mismo símbolo o equivalente) siempre que ambas piezas tengan uno de sus laterales libres, es decir, no tengan piezas a su derecha y a su izquierda al mismo tiempo. Por ejemplo, en la imagen siguiente podemos ver una pieza "W" seleccionada (en blanco sobre negro) con el lateral derecho libre (está sobre otra pieza), mientras que el cursor se dirige a la pieza "W" de la parte inferior izquierda del tablero, que tiene la parte izquierda de la pieza libre. Al seleccionar ambas piezas, éstas desaparecen del tablero, quedando 2 piezas menos, y dejando huecos gracias a los cuales podemos ahora seleccionar las piezas que lindaban con ellas. Nótese que no se podría haber seleccionado la "W" de la segunda fila por abajo, ya que ésta está "encerrada" entre la "N" y la "C", y por lo tanto no puede ser eliminada.

Para seleccionar las diferentes piezas movemos por pantalla un cursor con forma de flecha, utilizando las teclas o el Joystick (que puede ser Cursor, Kempston o Sinclair). Las teclas por defecto para controlar el juego son O, P, Q, A y Space (aunque podemos jugar con 6, 7, 8, 9, 0 si se selecciona la opción de Sinclair Joystick).



Eliminando piezas del tablero

En la siguiente captura podemos ver cómo hemos eliminado gran parte de las piezas (después de muchos minutos de juego) y apenas nos quedan 30 fichas para ganar (si conseguimos vaciar la pantalla en su totalidad) o perder (si jugamos las fichas incorrectas y nos quedamos en una situación en la cual no hay ninguna jugada posible que nos permita eliminar pieza alguna).



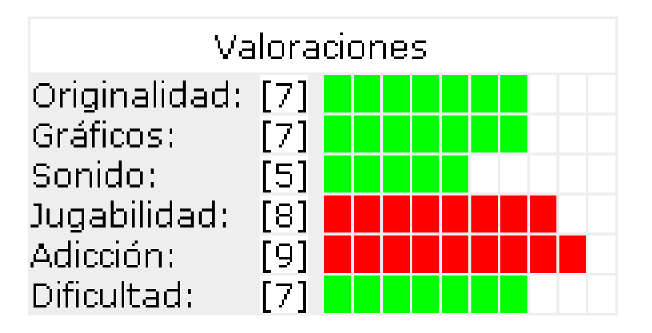
¿A punto de ganar la partida?

Un detalle a favor del juego son las diferentes opciones que permite mientras lo jugamos. Si pulsamos en el icono en forma de pergamino se desplegará un menú que nos permitirá:

- Exit Menú: Salir de menú y volver al juego.
- List tiles: Muestra una lista de todas las fichas de Mahjonng.
- Show Moves: Muestra todos los movimientos posibles en el momento de juego actual (alternando entre uno y otro mediante nuestra pulsación del botón de disparo).
- Change View: Permite visualizar el tablero de juego desde otra perspectiva (para poder ver piezas en posiciones en la que nos sea complicado ver el dibujo).
- Look Under: Permite ver todas las capas del tablero.
- Back / Forward: Deshacer y rehacer movimientos.
- Pause y Quit Game: Pausar o salir del juego.

Los gráficos del juego son bastante buenos, aunque el sistema de control puede ser un tanto complicado para mover la escurridiza flecha inicialmente. No existe música durante el juego, sólo durante el menú principal, y los efectos se reducen a un par de sonidos al destapar piezas correctas o intentar destapar piezas incorrectas. Pese a estas pequeñas deficiencias, la adicción al juego es máxima.

En resumen: un juego ideal para los amantes del Mahjonng, que puede dar para muchas horas de diversión debido a la aleatoriedad de este antiguo juego oriental. Es una verdadera lástima que no fuera publicado de forma comercial, aunque esto puede ser debido a que apareció en 1991, uno de los últimos años de "vida" del Spectrum. Un juego recomendado para pasar muchas horas con él.



Valoraciones	
Originalidad:	[7]
Gráficos:	[7]
Sonido:	[5]
Jugabilidad:	[8]
Adicción:	[9]
Dificultad:	[7]

Trucos:

POKE 40104,0 + POKE 51329,0 : Tiempo infinito.
POKE 51094,0 : Muestra de jugadas infinita.
POKE 39672,204 + POKE 39673,200 + POKE 39674,155 + POKE 39676,134 + POKE 39699,56 : Seguir viendo la pantalla durante la pausa.

Cargador:

(Por Hackers Anonymous Gerard Sweeney, Jamie Murray y Jon North en Your Sinclair nº 76)

```
10  REM Peking by Hackers Anon
20  CLEAR 64999: LOAD
    ""CODE
30  POKE 65173,48: POKE 65174,242
40  LET A=62E3
50  READ B: IF B=999 THEN RANDOMIZE
    USR 65146
60  POKE A,B: LET A=A+1: GO TO 50
70  DATA 175,50,243,145,50,94,189:
    REM INFY TIME
```

```
 80 DATA 175,50,115,118: REM NO TIME
    ADDED ON SHOW-MOVES
 90 DATA 33,204,200,34,67,144,62,155,
    50,69,144,62,134,50,71,144,62,56,
    50,94,144: REM NO BLACKOUT WHEN
    PAUSED
100 DATA 195,120,93,999: REM END
    MARKER
```

MIGHTY MAGUS



En un bello reino llamado Sauria habitan sabios, bellas damiselas, bravos caballeros... y Fraugy el Fiero, un temible Dragón que arrasa por las noches las aldeas y se come a los pobres campesinos... tras lo cual regresa a su guarida en las montañas: el Templo del Sol Naciente, consagrado al Dios Dragón. Lógicamente el rey no esta dispuesto a quedarse cruzado de brazos y ha acudido al mago Devlin para que ponga fin a tan molesta situación, dado que harán falta más artimañas que espadas para acabar con el Dragón. Lamentablemente Devlin esta muy ocupado observando otros aconteceres cósmicos y tendrás que ser tú, su humilde discípulo, quien deba enfrentarse al Dragón.

Mighty Magus es un curioso plataformas con scroll multidireccional con algunos toques que ilustran perfectamente lo que sería una concepción muy sencilla de un arcade ambientado en la típica mazmorra repleta de enemigos y trampas.

Nuestro objetivo en el juego como el hábil discipulo de mago, será llegar al nivel más profundo de la mazmorra de Fraugy, acabar con el Dragón y escapar de la mazmorra. Para ello tendremos que descender por los 30 niveles de los que consta la mazmorra, evitando trampas y el creciente número de enemigos, aprovechando los pocos recursos a nuestro alcance y la ayuda de nuestro fiel cayado mágico.



Un bonito unicornio al que más vale no acercarse...

En nuestro camino, a traves de las plataformas que conforman cada nivel, encontraremos cofres de tesoros y montículos de desechos que tanto pueden contener pociones para aumentar nuestros puntos de Magia, Poder, Vidas u otros objetos como un escudo mágico que nos protege durante un tiempo (el borde se pondrá amarillo durante ese intervalo). En otras ocasiones el buscar nos hará disparar otras trampas. Asimismo en cada planta de la mazmorra existe un gran número de trampas que se activan al

pisar, desde llamaradas a flechazos pasando por paredes que nos aplastarán con sus afilados clavos o teletransportadores que nos pueden hacer tanto bajar como subir a otro nivel de la mazmorra. Afortunadamente perder una vida no significará volver al inicio de la mazmorra sino a una de las entradas de ese nivel.

Gráficamente es muy sencillo, aunque aceptable para ser un juego del año 1985. El scroll es por bloques y por tanto ligeramente brusco sin llegar resultar molesto. Existe una opción para reducir el número de colores en el área de juego para los que usen monitores monocromos, televisores en blanco y negro, o sufran de epilepsia. Anecdóticamente también existe la opción de salvar y cargar la tabla de puntuaciones.

El sonido se reduce a una musiquilla en el menú y a los efectos sonoros al usar el bastón mágico o caer en una trampa... es poco pero ¿quién dijo que hubiese música de fondo en una mazmorra?

Las teclas son redefinibles, contando con las direcciones izquierda y derecha, luchar, salto (en la dirección a la que miremos) y buscar (que sirve también para bajar a medias de una escalera o no subir una escalera cuando pasamos junto a ella).



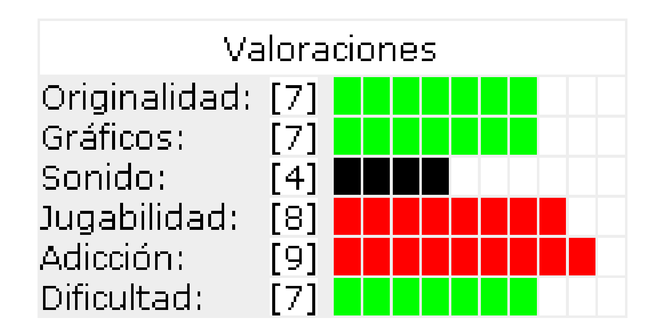
... a ese guardián no debió hacerle gracia que abriera ese cofre...

Es un juego original en su mezcla de conceptos que, pese a sus pequeños defectos gráficos, tiene ese toque adictivo que nos hace reintentar una y otra vez llegar al nivel más profundo de la mazmorra... a ver si derrotamos a ese Dragón, qué nuevos monstruos encontramos, a ver quién consigue más tesoros... después de todo resulta muy sencillo llegar casi a la mitad de la mazmorra... y eso nos incita a querer seguir un poco más cada vez.



¡No hay que dejar cofre por saquear!

Resumiendo: un desconocido pero entretenido juego de aventuras mágico-medievales al estilo plataformero con un diseño simpático y agradable para la época. Indudablemente depende de la primera impresión que nos cause debido a su factura técnica, pero sin duda es un juego con la capacidad de engancharnos durante bastantes horas.



Valoraciones	
Originalidad:	[7]
Gráficos:	[7]
Sonido:	[4]
Jugabilidad:	[8]
Adicción:	[9]
Dificultad:	[7]

Trucos:

A la hora de enfrentarse al Dragón, si permanecemos en las escaleras de su izquierda atacando con hechizos, ni el Dragón ni su fuego nos alcanzarán.

Otros Comentarios:

Crash issue 16:
ftp://ftp.worldofspectrum.org/pub/sinclair/magazines/Crash/Issue16/Pages/Crash160001 4.jpg

Sinclair User issue 39:
http://www.sincuser.f9.co.uk/039/sftwreb.htm

Your Spectrum issue 15:
http://www.users.globalnet.co.uk/~jg27paw4/yr15/yr15_51.htm



MIGUEL G. PRADA

Este mes vamos a pasar nuestros discos de Spectrum, tanto de 3" como de 3"1/2, a formato DSK para poder usarlos con un emulador o distribuirlos por medios digitales al resto de usuarios interesados, en lo que será la segunda y última entrega del análisis que hemos hecho al conjunto de utilidades que componen las CPDTOOLS.

Como complemento a todo esto explicaremos, con un ejemplo sencillo de comprender, cómo pasar un juego de cinta a disco. Esto vamos a hacerlo en un emulador de Spectrum corriendo sobre PC, pero es totalmente aplicable a un Spectrum real.

CONFIGURANDO CPDREAD

La instalación de las CPDTOOLS ya la abordamos en el anterior número de Magazine ZX, por lo cual vamos a pasar directamente a la parte práctica.

Lo primero que tenemos que hacer es abrir con un editor de textos el archivo CPDREAD.CFG y mirar si tenemos las opciones adecuadas seleccionadas. Aunque muchos de estos parámetros los podemos introducir por la línea de comandos, lo ideal es tener el archivo adecuado a nuestras necesidades.

En este archivo encontraremos una corta y sencilla serie de opciones que, pese a que se pueden dejar prácticamente como vienen, vamos a recorrer someramente para que podamos hacer nuestras variaciones si lo deseamos.

- #tracks

  Como su nombre indica es el número de pistas a leer del disquete, de momento podemos dejarlo como está, con valor 80.

- #drive

  La letra a la que está asignada nuestra unidad de disco de donde vamos a leer. Si lo hacemos de un disco de 3" 1/2 lo normal es que esté configurado como 'A', y si tenemos acoplada una unidad de 3" a nuestro PC, lo más lógico es tenerla como 'B'. Ponemos la letra que se corresponda con la unidad que queremos leer y ya está.

- #sides

  El número de caras que tiene el disco. Lo podemos dejar tal cual.

- #step

  Aquí se indica el número de 'tracks' que tiene que saltar el programa por cada lectura de la disquetera. Por defecto viene con un valor de '1', pero si vamos a leer desde la disquetera de 3", debemos cambiarlo por '2'. De lo contrario no saldrá el DSK correctamente.

- #mediatype

  El tipo de disquetera desde el que vamos a leer. Por defecto tiene el tipo '2' seleccionado, y con ése nos bastará tanto si es para la disquetera de 3" 1/2 cómo para la de 3". En el mismo archivo vienen el listado de disqueteras soportadas más la opción correspondiente que debemos poner.

Una vez configurado el archivo, salimos del editor grabando los cambios y ya tenemos el CPDREAD listo para su uso.

CREANDO NUESTRO DSK

Ahora viene la hora de crear nuestro primer DSK, tarea muy sencilla gracias a la cual

podemos disponer de una copia de seguridad de nuestros discos originales de Spectrum, o compartirlos mediante web o email con nuestros colegas; siempre en caso de que la casa propietaria de los derechos del juego no haya denegado su libre distribución y, valga como ejemplo de esto último, desarrolladores y distribuidores tales como Ultimate, Capcom, etc. De esto os podéis informar en **WOS**.

Como lo más normal es querer hacer la imagen de un disco de 3", adecuamos nuestro archivo de configuración a esta característica como indicamos en el paso anterior. Una vez hecho esto, deberemos arrancar nuestro PC en modo MS-DOS o bien utilizar un disco de arranque como indicábamos en la entrega del mes pasado. Introducimos el disco en la unidad de 3" conectada al PC y tecleamos:

`CPDREAD.EXE midisco.dsk` y pulsamos INTRO.

(Por supuesto en el directorio donde tengamos almacenado el programa, que bien puede ser 'C:/CPDTOOLS').

Después de unos segundos, y si nada falla, tendremos un archivo llamado 'midisco.dsk' con el contenido del disco por la cara correspondiente.

Para hacer la copia del segundo lado del disco le damos la vuelta, cambiamos el nombre del archivo, por ejemplo 'midisco_b.dsk' y ejecutamos la orden de nuevo.

Podemos encontrarnos con el caso de querer hacer un DSK de un disco de 3"1/2, grabado como si tuviese dos caras. Nada más sencillo. Volvemos a configurar el CPDREAD.CFG con las opciones idóneas y tecleamos:

`CPDREAD.EXE midisco.dsk` para la cara 'A' y

`CPDREAD.EXE midisco.dsk` S para la cara 'B', nótese la 'S' que indica al programa que lea de la cara 'B'.

Si no queremos modificar el archivo de configuración, podemos pasarle al programa los parámetros mediante la línea de comandos, por ejemplo:

```
CPDREAD.EXE midisco.dsk [tracks]
[drive]
```

En el cual le indicaremos el disco desde el que queremos leer de la misma manera que lo haríamos en el archivo de configuración. Esto queda a nuestra elección.

El programa, en caso de que no le pasemos ninguna opción por línea de comandos, las lee del archivo CFG, con lo cual prevalecen las primeras sobre las segundas.

CONCLUSIONES

El conjunto de herramientas que comprende el paquete CPDTOOLS nos va a ser muy útil para preservar nuestros discos o crearlos y, unido esto a la posibilidad de tener una disquetera de 3"1/2 conectada a nuestro Spectrum +3, se abre un amplio abanico de posibilidades.

Como habéis podido comprobar, tanto grabar como leer discos es muy sencillo, y únicamente nos encontramos con un pero bastante importante, que sólo funcione con MS-DOS como sistema operativo, que actualmente está obsoleto y poca gente lo tiene instalado en su PC. La falta de versiones para Linux o Windows le resta bastantes enteros.

Aún así es un pack de utilidades imprescindible si queremos tener una experiencia sensacional con nuestro Spectrum +3.

PASANDO UN JUEGO DE CINTA A DISCO

Cuando adquirimos un Spectrum +3, el principal motivo por el que lo hacemos es para evitar los largos tiempos de espera y los errores en la carga de los programas desde un soporte como es la cinta de cassette, poco fiable y que depende de varios parámetros para su correcta carga, como pueden ser el volumen, la colocación de la cabeza lectora del reproductor de audio mediante el azimut (ese tornillito pequeño que tenemos que girar para hacer subir o bajar la cabeza lectora respecto de la cinta), o el estado de la cinta misma (y la suerte, por qué no decirlo).

A partir del lanzamiento de este modelo muchas empresas creadoras de software sacaron sus productos tanto en cinta de cassette como en disco de 3". Pero el software anterior, salvo contados packs recopilatorios, no lo tenemos disponible en disco.

Este es el motivo de la creación de este artículo: pasar nuestros juegos preferidos de cinta a disco.

ALGUNAS CONSIDERACIONES PREVIAS

Los programas almacenados en cinta de cassette, en muchos casos, estaban puestos a salvo de la piratería y la copia indiscriminada con diferentes sistemas de protección por software. Carga turbo, eliminación de las cabeceras de información, grabación de la pantalla, basic y código máquina como un solo bloque, y un largo etcétera.

Este texto no pretende ser un manual de cómo desproteger programas, con lo que, en esta ocasión, nos vamos a limitar a los que utilizaban la rutina de carga de la ROM con los comandos típicos del BASIC: 'LOAD' y 'SAVE'.

Antes de decidirnos a pasar un programa a disco, debemos asegurarnos de que mantiene compatibilidad con el Spectrum +3. Varios juegos no funcionan en este modelo de Amstrad o hay que modificarlos para que lo hagan. Por lo cual la mejor manera de asegurarse si nos van a funcionar es cargándolos en el mismo, bien sea un TZX o TAP en el emulador o la cinta original en el +3.

Por comodidad, en vez de trabajar sobre la máquina real, lo haremos sobre un emulador. En concreto vamos a usar FUSE para Linux, ya que es el que el autor usa habitualmente. No obstante, resulta válido cualquier otro que soporte el uso de archivos DSK, como imágenes de disco, y TZX o TAP como cintas, en cualquier plataforma o sistema operativo (MS-DOS, Windows, etc.). Incluso se puede seguir el artículo trabajando directamente el +3 (eso sí, las cargas de cinta se harán más tediosas).

INFORMÁNDONOS DE CÓMO ESTA GRABADO UN PROGRAMA

Los programas grabados con los comandos habituales del basic 'LOAD' y 'SAVE' tienen un esquema estándar. Cuando ejecutamos el SAVE "programa" LINE 10, primero se graba una cabecera con la información que necesita la rutina de la ROM para cargar el bloque de datos, y después el bloque de datos propiamente dicho. Para pasar el programa a disco, necesitamos saber la información exacta, y ésta se puede conseguir con un programa lector de cabeceras. Podemos bajarnos un programa que publicó Microhobby en el número 2 con el nombre de 'Listador de cintas', que cumplirá con las funciones requeridas perfectamente, de la web KyENTER. Está grabado en un TAP gracias al webmaster de la misma y nos va a ser de muchísima utilidad.

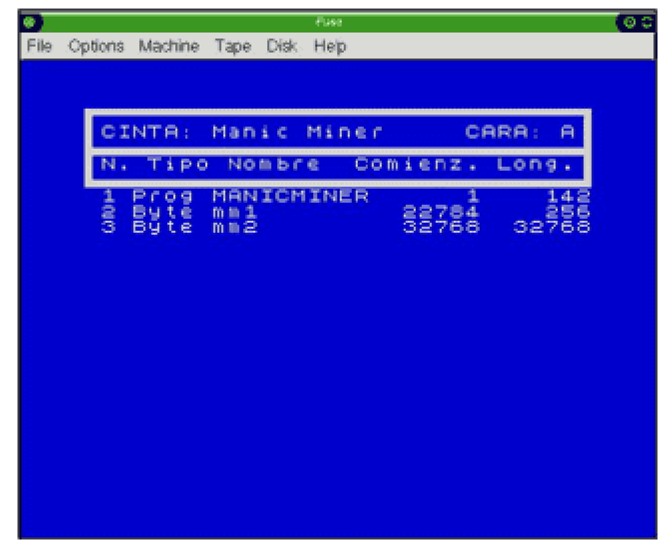


Imagen 1. El listador de cintas en funcionamiento

Vamos a elegir un clásico para pasar a disco, como es el Manic Miner. Es un juego muy sencillo de grabar en disco siguiendo las indicaciones aquí dadas.

Comenzamos por abrir nuestro emulador y quitar la opción de auto-carga de cintas en caso de que la tuviésemos. Debemos elegir al Spectrum +3 como modelo de hardware que queremos emular.

Abrimos el archivo del 'Listador de cintas' en el emulador y pasamos al BASIC con el menú del +3. Una vez en él tecleamos `LOAD "T:"` y ENTER, con esto conseguimos que el

+3 mande las ordenes de carga a la cinta de cassette y no a la unidad de disco que es a donde van por defecto en este modelo.

Tecleamos en esta ocasión `LOAD ""` y cargaremos el listador en memoria. Una vez cargado, nos saldrá una pantalla azul que nos pregunta por el nombre de la cinta, podemos ponerlo o simplemente dar ENTER, esta información vale realmente para poco; una pregunta similar nos hará respecto a la cara de la cinta y tiene el mismo valor que la anterior. Antes de pulsar ENTER de nuevo, abrimos el archivo TAP o TZX del que queremos sacar la información, en nuestro caso el que lleva el juego Manic Miner, que hemos bajado de **WOS**. Pulsamos ENTER y al cabo de un par de segundos nos saldrá una pantalla como la de la Figura 1 con parte de la información que necesitamos.

Podemos ver que el Manic Miner está grabado con tres bloques de datos:

1. BASIC con nombre 'MANICMINER' que se autoejecuta en la línea 1 y tiene una longitud de 142 bytes.
2. Bytes con nombre 'mm1' que comienza en la dirección 22784 y tiene una longitud de 256 bytes.
3. Bytes con nombre 'mm2' con comienzo en 32768 y una longitud de 32879 bytes.

MODIFICANDO EL CARGADOR BASIC PARA EL +3

Con esta información ya podemos pasar el programa a disco, pero antes tenemos que modificar el cargador BASIC para adecuarlo a nuestro +3.

Reiniciamos el Spectrum (el emulador en este caso) y, desde el BASIC, volvemos a teclear `LOAD "T:"` y ENTER, y a continuación `MERGE ""`. Así cargaremos en memoria el cargador del Manic Miner, sin que llegue a ejecutarse gracias al MERGE.

Miramos el listado BASIC (que tenemos en la Figura 2) del mismo y es importante que tomemos nota de dos cosas: la dirección del CLEAR y la del USR, que en nuestro caso vienen a ser 24310 para el CLEAR y 33792 para el USR. La primera dirección nos indica el lugar donde colocamos el RAMPTOP para que el BASIC no machaque la información y la PILA del ordenador, y la segunda nos indica la dirección de arranque del programa en código máquina.

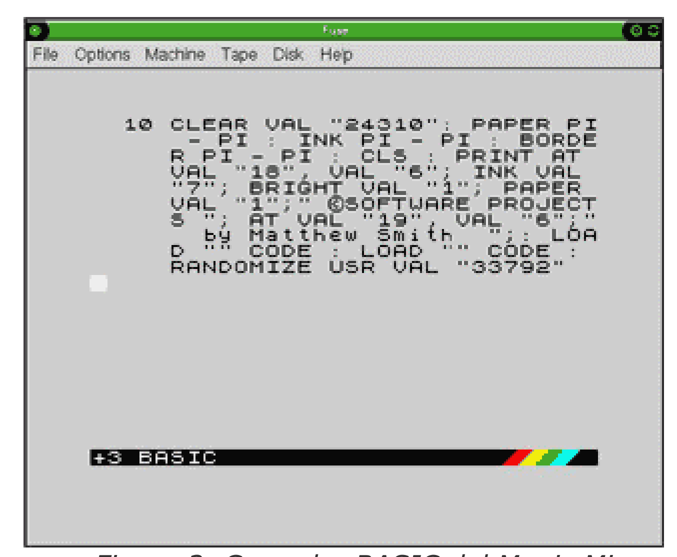


Figura 2. Cargador BASIC del Manic Miner

Cuando cargamos un programa de cinta no es necesario indicar el nombre de los bloques que se van a cargar, ya que se supone que van grabados secuencialmente en el mismo orden que los vamos a cargar. Esto no pasa con el +3 y la unidad de disco, debemos indicarle exactamente cómo se llaman los diferentes bloques, en caso contrario no sabría localizarlos. Los nombres que podemos dar a los diferentes archivos en la unidad de disco cambia respecto a como estamos acostumbrados a hacerlo en cinta. En disco se pueden nombrar con el formato de 8+3, es decir, ocho caracteres para el nombre, seguidos de un punto, y tres caracteres para la extensión que nos indique de que tipo de datos está compuesto el archivo. Por ejemplo "manic.bas" para el cargador basic del Manic Miner, "manic.scr" para la pantalla de carga, "manic.bin" para el bloque de datos y código máquina. En cambio, a la hora de nombrar programas en cinta se pueden emplear hasta un total de 10 caracteres, sin extensiones, a nuestra entera disposición.

Con el cargador en memoria tenemos dos opciones, modificar el que tiene el juego original o crear uno nuevo. En este caso simplemente vamos a poner los nombres que deseamos en las órdenes LOAD; los mismos con los que vamos a grabar los bloques posteriormente a disco. El cargador debe quedar como en la Figura 3. Veréis que

tiene los comandos tecleados de una forma un tanto especial, con instrucciones tales como VAL, PI, etc. Esto es así para que el programa BASIC ocupe menos en memoria.

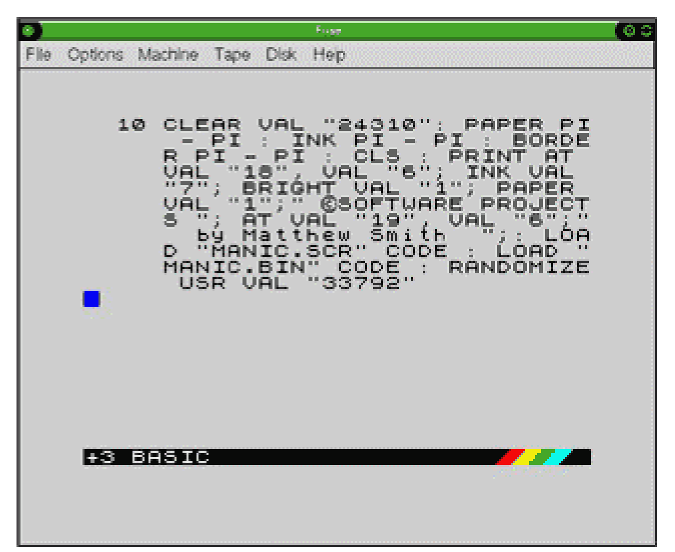


Figura 3. Cargador para la versión en disco

GRABANDO EL PROGRAMA EN EL DISCO

Al hablar de disco, nos referimos indistintamente a un archivo DSK cargado en el emulador (para lo cual debemos conseguir un DSK en blanco) o bien un disco virgen en el +3. Para el emulador, podemos coger el archivo 180k.dsk que viene con las CPDTOOLS y hacer una copia con otro nombre.

Abrimos el DSK (vacío) en el emulador y empezamos a grabar los diferentes bloques. El cargador BASIC lo tenemos en memoria, por lo cual tecleamos SAVE "manic.bas" LINE 10 y pulsamos ENTER. Esperamos unos segundos, y estará grabado en el disco. Los dos bloques de datos deberemos cargarlos ya que no lo hicimos antes.

Ejecutamos `CLEAR 3E4` y `ENTER`, por ejemplo, para evitar que se nos corrompa el bloque que vamos a cargar en 32768 y a continuación `LOAD "" CODE: SAVE "manic.scr" CODE 22784,256` y pulsamos ENTER. Con esto cargamos de cinta y despues grabamos a disco el primero de los bloques. Con el segundo hacemos algo similar: `LOAD "" CODE: SAVE "manic.bin" CODE 32768,32768` y ENTER. Ya tenemos el programa grabado en disco. Para comprobarlo hacemos `CAT` y ENTER y debería salirnos algo similar a lo mostrado en la Figura 4.

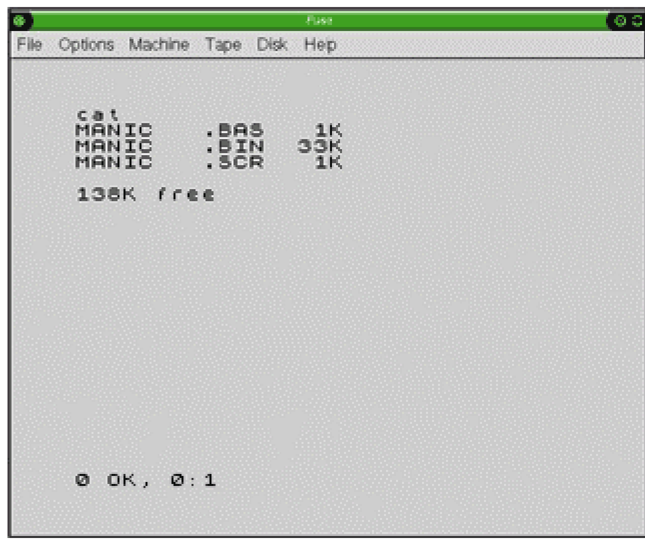


Figura 4. Listado del disco que acabamos de hacer

Ya tenemos el Manic Miner listo para ser cargado en segundos en nuestro +3. Reseteamos el emulador y tecleamos LOAD "manic.bas" y ENTER. Al cabo de unos segundos, y si seguimos el proceso aquí explicado, tendremos el juego cargado y listo para pegarnos unas partiditas.

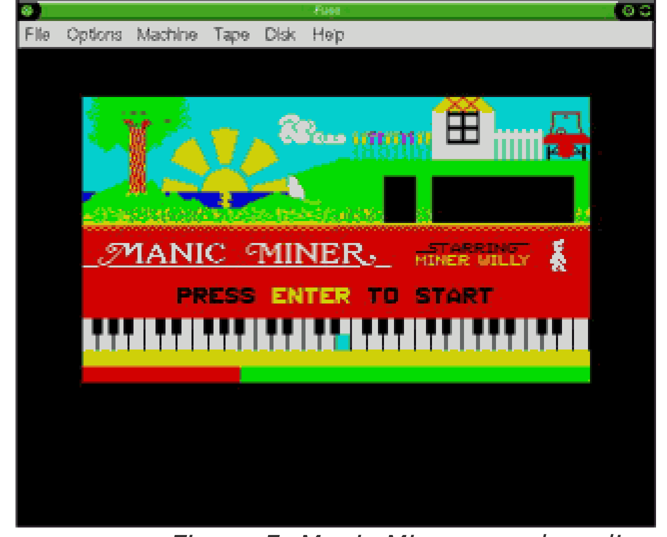


Figura 5. Manic Miner pasado a disco

CONCLUSIONES

Por supuesto, todo esto que acabamos de realizar lo podemos hacer sobre un +3 real, utilizando cintas y discos reales, etc., con el inconveniente del tiempo de carga. El emulador nos facilita enormemente esta tarea y, además, no olvidemos que podemos grabar el archivo DSK generado a un disco de 3" o 3"1/2 como explicábamos en esta misma sección el mes pasado.



FEDERICO ÁLVAREZ

ADAPTADOR PARA DISQUETERA DE 3 ½ PARA EL +3 (y II)

Continuamos un mes más nuestra sección de hardware completando la serie iniciada en el primer número de Magazine ZX, dedicada a conectar una disquetera de 3"1/2 al Spectrum +3.

El mes pasado dejamos la disquetera alimentada y conectada al Spectrum, con lo cual habremos podido empezar a cacharrear y disfrutar de ella. Sin embargo, necesitamos hacer algunos ajustes adicionales para poder acceder a toda su funcionalidad. En concreto, vamos a añadir dos interruptores al cable de datos. El primero de ellos nos permitirá configurar la disquetera como primera unidad (A:) o segunda unidad (B:). El segundo de ellos nos dará acceso a usar ambas caras del disco. Veámoslo en profundidad.

MATERIALES

Con esta parte del montaje nuestra economía no sufrirá demasiado, ya que no necesitamos muchas cosas.

- 2 interruptores (en nuestro caso los emplearemos dobles y de 3 patillas, ya que disponíamos de algunos), pero usaremos únicamente 2 de ellas.
- Cuatro trocitos de alambre o cable pelado, de aproximadamente medio centímetro de longitud cada uno (en caso de que hagamos el montaje a través del conector del cable como se explica en el artículo, sin soldaduras).
- Material para soldar (soldador, estaño y cable), en caso de que optemos por cortar y empalmar cables para realizar el montaje).

CONFIGURANDO LA UNIDAD COMO PRIMARIA O SECUNDARIA

Tal y cómo está el montaje, la disquetera queda configurada como secundaria (B:). Podemos comprobarlo haciendo CAT "B:" desde el BASIC del +3, y obtendremos un listado con el contenido del disco de 3"1/2.

Para configurarla como primaria (A:) debemos hacer un puente entre los pines 23 y 24. Esta puede ser una buena solución si nuestra disquetera original de 3" está averiada o no pretendemos hacer uso de ella. Sin embargo, para poder emplear ambas cuando sea necesario, es mejor instalar un interruptor. Nosotros hemos empleado interruptores como los de la Imagen 1.

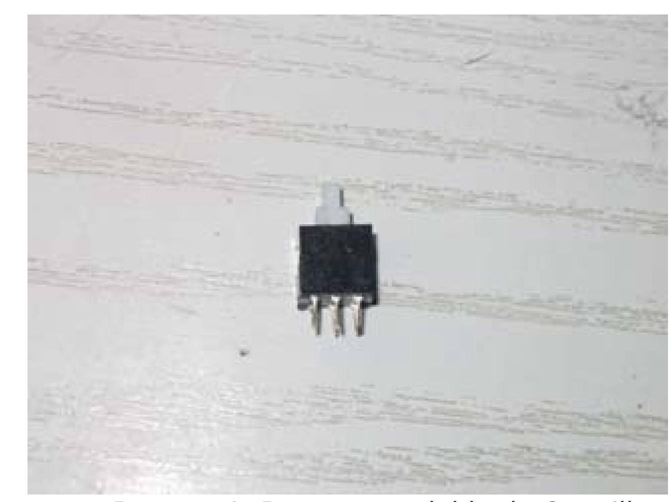
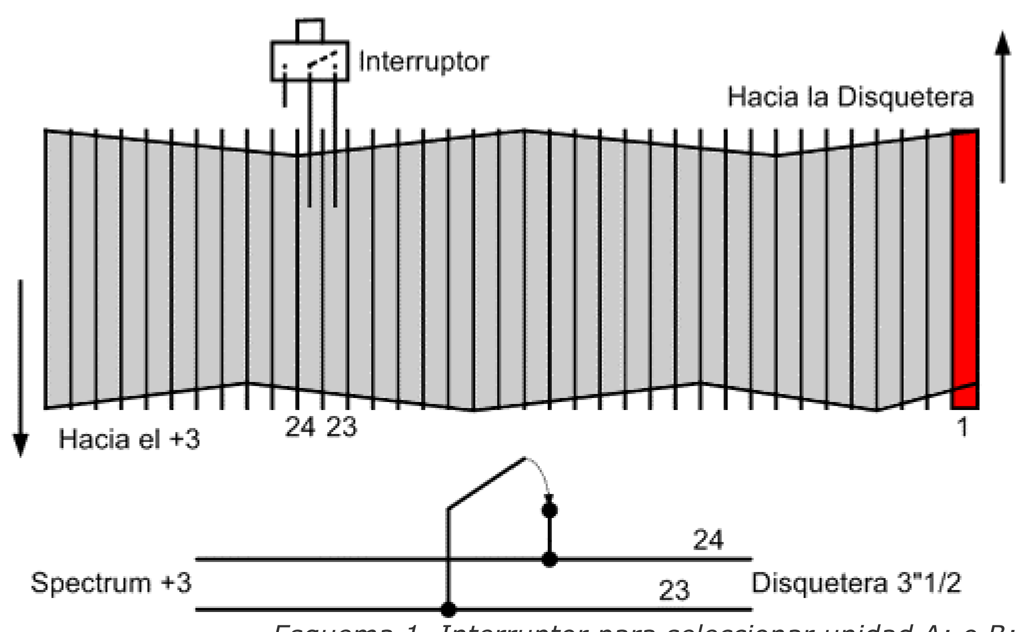


Imagen 1. Interruptor doble de 6 patillas

Siguiendo con la táctica empleada en la entrega anterior, haremos el montaje en uno de los conectores de disquetera de 3"1/2 de los que dispone el cable, ahorrándonos cortar cables, empalmarlos y soldar. No obstante, cada cual puede efectuarlo de la manera que considere más cómoda, más apropiada o estéticamente más elegante. La cuestión es seguir el esquema eléctrico teórico (Esquema 1).



Esquema 1. Interruptor para seleccionar unidad A: o B:

En nuestro caso concreto hemos introducido directamente las patillas del interruptor en los orificios del conector.

Para que no se produzcan fallos de conectividad, y sin necesidad de soldar nada, basta con introducir un fragmento de cable pelado fino o alambre para que aumente la presión entre las patas del interruptor y el conector, tal y como se muestra en la Imagen 2. De esta forma, al introducir el interruptor, no hay holgura y queda mejor fijado el montaje.

Debido al tipo de interruptor usado y al montaje que estamos haciendo, en uno de los conectores libres, sin modificar el cable de disquetera original, es necesario doblar o cortar una patilla del interruptor (una de las que caigan en los pines 27 ó 28) para no efectuar un corto entre dichos hilos.



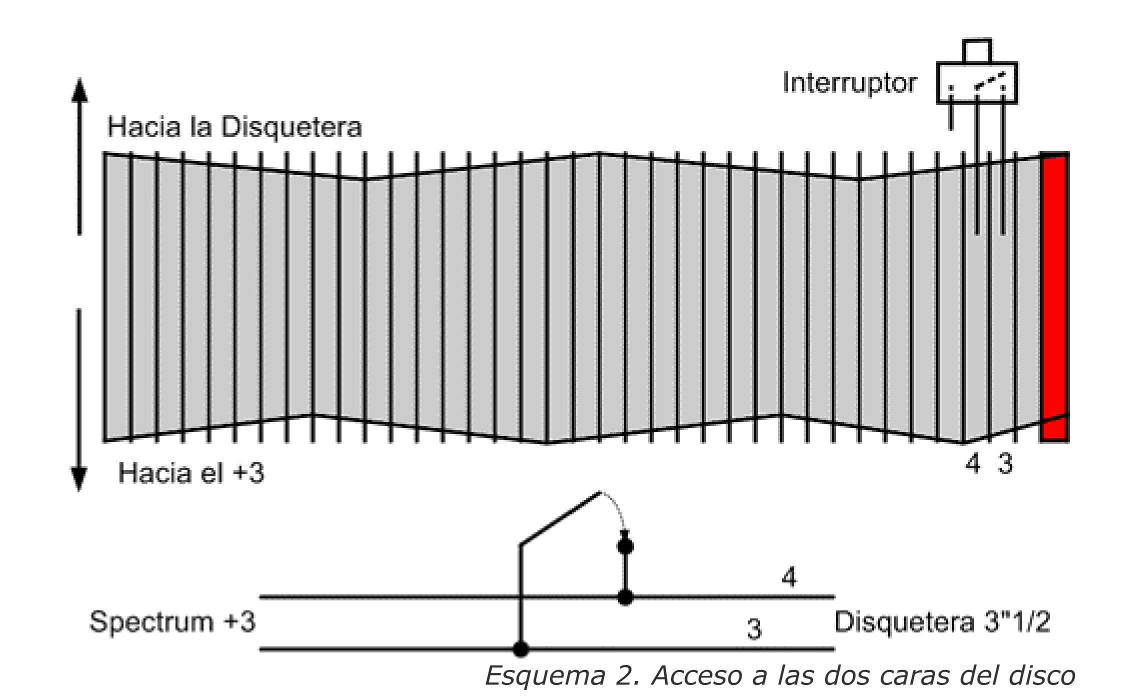
Imagen 2. Alambres en los conectores para mejorar la conectividad (pines 3, 4, 23 y 24)

Al cerrarse el circuito, la disquetera de 3"1/2 será reconocida como unidad "A:", mientras que al abrirlo se comportará como unidad "B:". Si bien este cambio se puede hacer con el Spectrum encendido, no recomendamos hacerlo, simplemente por precaución.

ACCEDIENDO A LA SEGUNDA CARA DEL DISCO

Para cambiar de cara de lectura-escritura, debemos instalar un interruptor entre los pines 3 y 4. De esta forma, al dejarlos desconectados accederemos a la primera cara del disco. Si los conectamos, accederemos a la segunda.

Emplearemos un interruptor idéntico al usado en el apartado anterior, siguiendo el Esquema 2. La forma de realizar físicamente el montaje es idéntica al que acabamos de hacer para seleccionar la configuración de la disquetera como primera o segunda unidad. Igualmente, será necesario doblar o cortar alguna de las patillas que caigan en los pines 7 u 8 para evitar cortocircuitarlos.



Esquema 2. Acceso a las dos caras del disco

De esta manera, el gesto de darle la vuelta al disco se sustituye por una simple pulsación en el interruptor que acabamos de instalar. Por descontado que se puede realizar con el Spectrum encendido, ya que no implica ningún riesgo. En realidad, estamos simulando la generación de una señal de control (seleccionar una u otra cara) que el Spectrum no sabe comunicar (ya que su disquetera no cuenta con dos cabezas de lectura).

Como resultado obtenemos el cable que se ve en la Imagen 3.

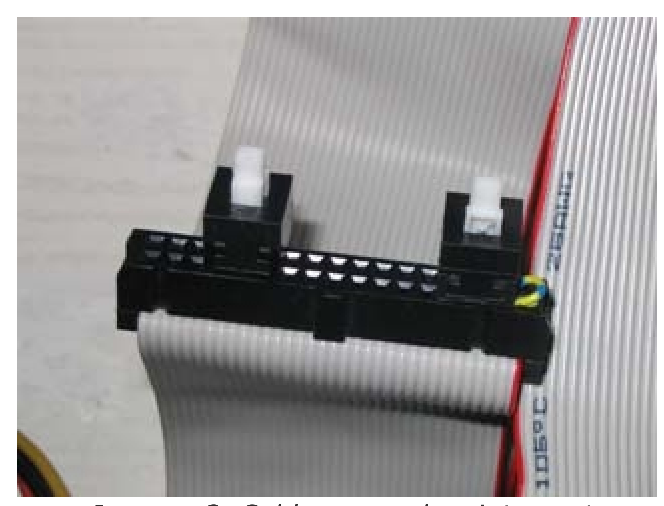


Imagen 3. Cable con ambos interruptores conectados

CONSIDERACIONES FINALES

En cualquier caso no se detecta la presencia o ausencia de discos, ya que hemos puenteado los pines 1 y 2. La disquetera estará en estado siempre

preparado, e intentará leer cuando le llegue la orden de hacerlo desde el Spectrum. Es por ello que el led de la disquetera de 3"1/2 siempre permanece encendido.

En caso de que tengamos ambas disqueteras (interna y externa) operativas, y la externa esté configurada como primaria, tendrá prioridad ante la secundaria. Por ejemplo, si tenemos un disco válido en la unidad de 3" y la unidad de 3"1/2 vacía, al seleccionar la opción 'Cargador' del menú de arranque del +3, el ordenador se quedará a la espera de poder leer un disco en la unidad externa.

No todas las disqueteras de 3"1/2 son válidas para este montaje. Lo recomendable es hacernos con una que sea lo más antigua posible. Las nuevas vienen configuradas por defecto como unidades secundarias, y hay que hacer modificaciones en la placa de la disquetera que, si bien no son complicadas, son particulares para cada marca y modelo y exceden el ámbito de este artículo.

Este mismo montaje es válido para los ordenadores Amstrad CPC6128. Por tanto, sin hacer ninguna modificación (salvo lo relacionado con la alimentación de la propia disquetera), se puede emplear la unidad externa indistintamente en ambos ordenadores.

Y simplemente desear que disfrutéis de la posibilidad de acceder a un montón de software ahorrándoos la tediosa espera que impone cargar desde la cinta. Nos despedimos hasta el próximo número en el que presentaremos otro montaje que creemos bastante interesante.

LINKS

- Cómo conectar una disquetera 3 1/2" al +3:
  http://www.speccy.org/sinclairmania/disquetera_PC_al_plus3.html
- Conexión de una unidad de 3.5”” al CPC 6128:
  http://www.terra.es/personal/diegovp/pag2.htm
- Colección de imágenes de discos en SPA2:
  http://www.speccy.org/spa2/spanish/disks.htm



SANTIAGO ROMERO

INTRODUCCIÓN A LA EMULACIÓN (I)

Algunos de nuestros lectores nos han expresado una serie de dudas relacionadas con la emulación, especialmente dudas de base y de concepto. Estas mismas personas nos han comentado la posibilidad de que desde Magazine ZX se muestre a la gente nueva en el mundo de la emulación un tutorial sencillo que explique qué es y cómo funciona un emulador, focalizándolo finalmente en la emulación de Spectrum.



Los tiempos cambian y resulta cada vez más difícil encontrar en el mercado aquellos viejos ordenadores, consolas o máquinas recreativas que utilizábamos hace algunos años. En algunos casos en que todavía están disponibles, a veces lo son a precios muy elevados, como en el caso de ordenadores Sinclair Spectrum +3 o QL, por ejemplo. En otros casos los precios pueden ser desorbitados, como los de las consolas NOMAD de SEGA, o ciertas máquinas recreativas. También existen sistemas antiguos muy baratos, pero cada vez quedan menos (debido a averías, o consolas y ordenadores que acaban en los contenedores de basura por usuarios que no las valoran) y llegará un momento en que será realmente complicado conseguirlos.



Figura 2. Sinclair ZX81: difícil de encontrar

No hay nada mejor que tener la máquina original y software de la misma para divertirse con ella, pero en ocasiones nos puede resultar imposible.

Los emuladores son programas destinados a recrear arquitecturas de ordenadores, consolas o recreativas en un determinado Sistema. Por ejemplo, existen emuladores que nos permiten jugar a juegos de Spectrum (o de otros ordenadores antiguos) en nuestro ordenador actual. Existen también emuladores de las diferentes consolas clásicas, de forma que podemos jugar a la Nintendo NES o a la SEGA Megadrive en nuestro ordenador, y cargar en ellos los juegos tal y como eran en cartucho, disco o cinta.

Los emuladores nacieron por diversos motivos:

- Como reto para los programadores, que querían hacer programas capaces de recrear o emular una determinada máquina antigua en una moderna.
- Para poder jugar y utilizar arquitecturas antiguas que no tenemos a nuestro alcance.
- Con el fin de preservar todo el legado de estas arquitecturas, de forma que cuando estas ya no estén disponibles, todavía tengamos emuladores que nos permitan ver todo su catálogo de software.
- Para desarrollar en estas consolas y ordenadores, trabajando en nuestro ordenador habitual y probándolo en un emulador. Con esto nos evitamos grabar el juego en un cartucho, disco o cinta para tener que probarlo en el sistema real.

Qué es un emulador

Un emulador es un programa destinado a recrear internamente el funcionamiento de una arquitectura diferente a aquella en que se ejecuta. El emulador no es más que un programa, sin partes hardware, que utilizando los recursos de la máquina donde se ejecuta, simula el comportamiento de la CPU, memoria y demás elementos de una máquina determinada.

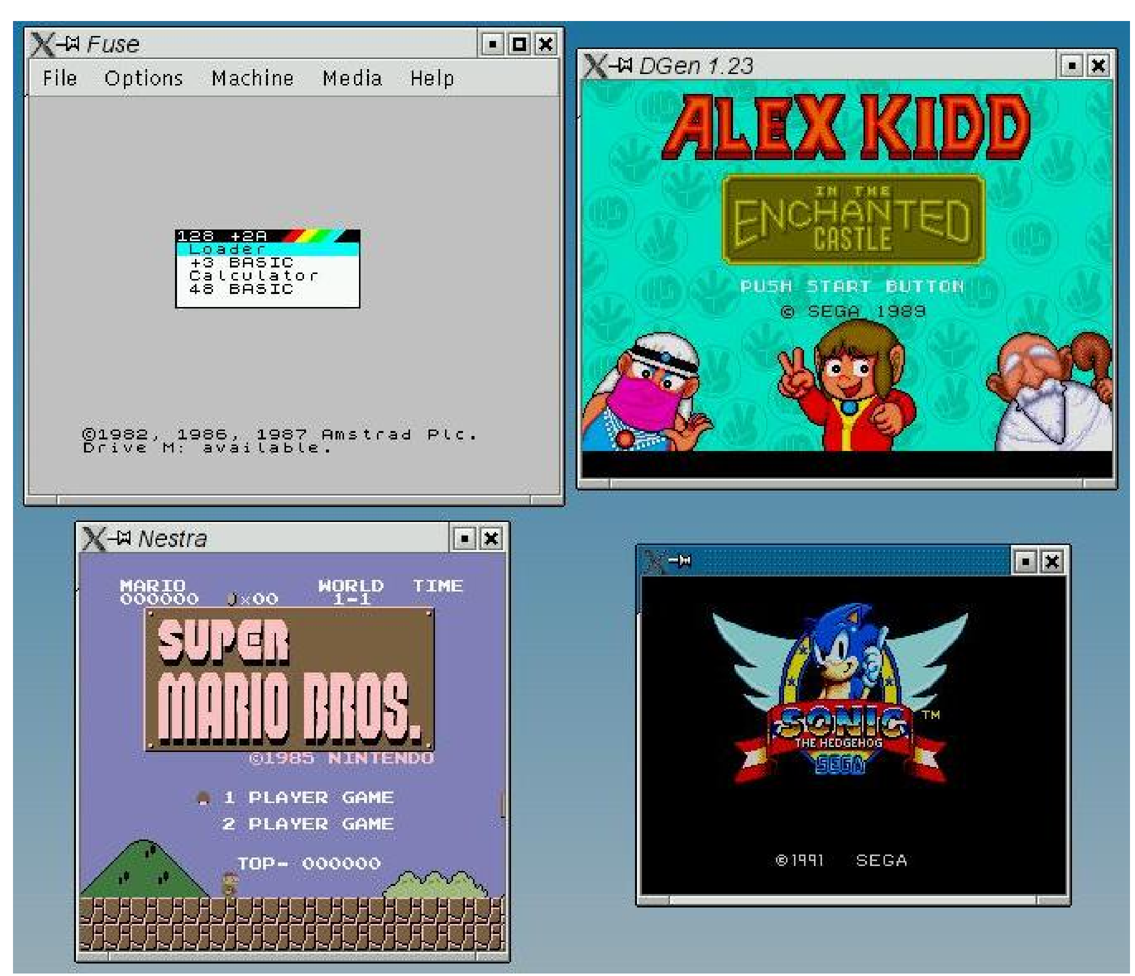


Figura 3. Emuladores de Spectrum, Megadrive, NES y Master System

Por ejemplo, Realspectrum es un emulador de Spectrum, ya que es un programa diseñado para emular un micro Z80 (que es el micro del Spectrum), una ULA (que es el "chip gráfico" del Spectrum), una unidad de cinta o de disco, etc... Del mismo modo que existe Realspectrum, NESticle es un emulador de NES que emula el micro de la NES, los chips de la NES, es capaz de ejecutar juegos en cartucho de NES, etc.

Normalmente los emuladores son programas que simulan una única arquitectura, aunque en algunos casos existen "macroemuladores" capaces de emular varios sistemas, como por ejemplo MAME (Multi Arcade Machine Emulator) que emula gran variedad de microprocesadores diferentes, lo cual le permite recrear gran cantidad de máquinas recreativas.

Nótese un detalle muy importante: un emulador no es más que un programa como cualquier otro instalado en la máquina. La diferencia está en que es vez de editar un texto, hacer cálculos, dibujar, o jugar (como el resto de programas), lo que hace es comportarse tal y como lo haría el sistema emulado. Posteriormente veremos con más detalle cómo.

Arquitecturas emuladas

Actualmente hay muchísimos sistemas emulados. Veamos algunos de los más conocidos (no están todos, pero sirven para hacernos una idea).

- Consolas:
  Nintendo: Nintendo NES, SuperNES, GameBoy, Nintendo64...
  Sega: GameGear, Master System, Megadrive/Genesis, Saturn...
  Atari: VCS 2600, 7800, Lynx...
  Otros: NeoGeo, PCEngine/TurboGrafx...
- Máquinas recreativas:
  Emuladas casi el 100% de ellas (salvo excepciones en determinados sistemas) por MAME (MultiArcade Machine Emulator) o RAINE.
- Microordenadores y ordenadores de 16 bits:
  Spectrum, Amstrad, MSX, Commodore, Atari ST, Amiga...
- Otros:
  Consolas de una y dos pantallas HandHeld y Game&Watch, Máquinas de pinball, etc...

Si buscas un emulador que te permita emular alguno de estos sistemas, puedes encontrarlo fácilmente en Emulatronia, pulsando en la sección correspondiente en la barra de menú de la izquierda.

¡Ojo! En algunas circunstancias puede ser ilegal descargar juegos de arquitecturas que todavía están a la venta, ya que al usarlas estás evitando al compra de la máquina y del juego. Es por ello que en algunas Webs de ROMS para emuladores encontrarás letreros indicando que sólo puedes descargar la ROM si eres el propietario del juego original, o bien que sólo puedes probarlas 24 horas, y condiciones similares. Quede claro que desde Magazine ZX estamos a favor de un uso correcto de la emulación: siempre hay que emular sistemas permitidos, usando ROMs cuya distribución haya sido permitida. Hacer lo contrario es perjudicial para el mundillo de la emulación, como veremos en la sección de Ventajas y Desventajas.

Debe quedar claro que los emuladores no son exclusivos de la plataforma PC. Es decir, no sólo hay emuladores de los diferentes sistemas para usarlos en PCs. Existen emuladores de Spectrum para Amiga, emuladores de Atari 2600 para Atari ST, o emuladores de NES para Playstation, por ejemplo. Recordemos que el emulador no es más que un programa que puede realizarse para cualquier plataforma.

Cómo funciona un emulador

Y bien, se preguntará algún lector, ¿cómo es posible que se pueda emular una máquina? ¿Qué tipo de "magia" realiza el emulador para mostrar un Spectrum dentro de mi ordenador? Lo que vamos a ver a continuación es cómo funciona a nivel interno un emulador. El usuario realmente no necesita saberlo, porque para él basta con abrir el emulador, cargar un juego o programa dentro del mismo (con las opciones que tiene el emulador para seleccionar un fichero ROM, cinta o disco de

nuestro disco duro), y utilizarlo, pero nunca viene mal entender cómo funcionan las cosas, y si estás leyendo este artículo es, precisamente, porque estás interesado en aprender el cómo y porqué de la emulación. No debes preocuparte si no conoces o entiendes algún concepto (aunque para algunos, esto será una explicación básica): lo importante es quedarse con la idea general de lo que vamos a ver durante este apartado.

Lo que en realidad el programador crea es, a grandes rasgos, el emulador de un microprocesador. Por ejemplo, es posible hacer un programa que lea instrucciones del microprocesador Z80, las comprenda, las ejecute, y guarde los resultados de las ejecuciones.

Un emulador de Z80 podría ser un programa hecho en C, en Visual BASIC, en PASCAL o en ensamblador (por citar algunos lenguajes comunes) que entienda las instrucciones en código máquina del Z80 proporcionadas al emulador, las ejecute y cambie los registros emulados del microprocesador, quedando todos los registros emulados igual que quedarían en un Z80 real si ejecutara el mismo código. Así, se podría realizar un programa en código máquina de Z80, y al ejecutarlo en un emulador, o en un Z80 real, obtendríamos exactamente los mismos resultados en los diferentes registros del microprocesador. Lo que estamos haciendo es recrear el hardware del sistema.

Ahora bien, con esto tenemos programas capaces de emular el hardware de una máquina, pero estos programas en sí mismos sólo emulan un microprocesador, o un chip gráfico, y no son capaces de hacer nada por sí mismos. Sería como si cogiéramos un PC y le quitáramos el disco duro y el chip de la BIOS. Al encenderlo tendríamos la pantalla en negro, ya que el micro (que lo único que hace es ejecutar una instrucción tras otra) no tiene nada que ejecutar. Para que puedan realmente hacer algo, necesitan un Sistema Operativo (más o menos complejo). Este sistema suele ser un chip ROM (memoria de sólo lectura) con las funciones básicas grabadas en él.

Al arrancar un Spectrum, por ejemplo, lo que se tiene es una ROM de 16KB (16384 bytes) que contiene el arranque de la máquina, el intérprete de BASIC, y todas las funciones necesarias para cargar y ejecutar el software de Spectrum. Esta ROM en realidad es un chip de memoria con el Sistema Operativo del Spectrum (un programa escrito en código máquina de microprocesador Z80) grabado en él. Este programa no tiene diferencias prácticas con un juego o un programa en cinta o disco de Spectrum, simplemente que su función no es jugar, sino dotarnos de un interfaz de gestión del Spectrum (el BASIC) y que en lugar de estar grabado en una cinta, lo está en una memoria de sólo lectura. Es decir, sí: hubo una persona (o varias) que programó (como si fuera un juego) el arranque de la máquina, el menú, el BASIC, etc., y lo ensambló con un "programa ensamblador" para obtener el código máquina almacenado en esa memoria ROM.

Nada nos impediría programarnos nuestro propio S.O. para Spectrum, grabarlo en un chip ROM igual que el original (de 16KB), abrir el Spectrum y reemplazar el chip de memoria de la ROM por el nuestro. Al arrancar el ordenador (suponiendo que hayamos hecho un S.O. bien programado) tendríamos un ordenador nuevo, totalmente diferente, con su sistema operativo. No sería un Spectrum tal y como lo conocemos, sino un ordenador nuevo que podría comportarse como un Spectrum o no (según cómo hayamos programado su BIOS). Estaríamos usando el mismo hardware del Spectrum (un Z80, una ULA, etc.) pero al ser diferente el Sistema Operativo, el ordenador funcionaría de forma diferente. Es el mismo caso que un PC (Micro Intel, tarjeta gráfica determinada, etc.) al que le podemos poner Linux, o Windows, y se comportaría de forma diferente (siendo el mismo hardware). He ahí la importancia de la ROM del sistema: es la que define cómo se comporta el ordenador, la que le da identidad. Sin ella, el ordenador no haría nada. Esa ROM en el caso del Spectrum (y de muchos otros sistemas) se graba en un chip.

Nuestro querido Spectrum toma el contenido de ese chip y lo ejecuta, instrucción a instrucción, como si fuera un programa (que de hecho, es lo que es), y va realizando todas las acciones que éste le dicta: borrar la pantalla, imprimir el mensaje de "(c) 1982

Sinclair Ltd Research", esperar la pulsación de una tecla, interpretar nuestros comandos, etc.

El emulador, como el Spectrum real, realiza la misma función. Éste emula un micro Z80 (y los demás componentes del Spectrum), y lo primero que hace el emulador es cargar desde fichero (como por ejemplo, rom48k.rom) el contenido de esta ROM del Spectrum, de forma que el emulador, al igual que el Spectrum real, pueda arrancar, ejecutando las instrucciones de la ROM del Spectrum mediante su "microprocesador emulado". ¿Cómo se ha obtenido este fichero con la ROM? Sencillo: se coge un chip con la ROM del Spectrum, se inserta en un lector/grabador de memorias, y se lee, grabando el contenido de los 16384 bytes en un fichero. Cada byte de este fichero se podría decir que es una instrucción del programa de la ROM del Spectrum.

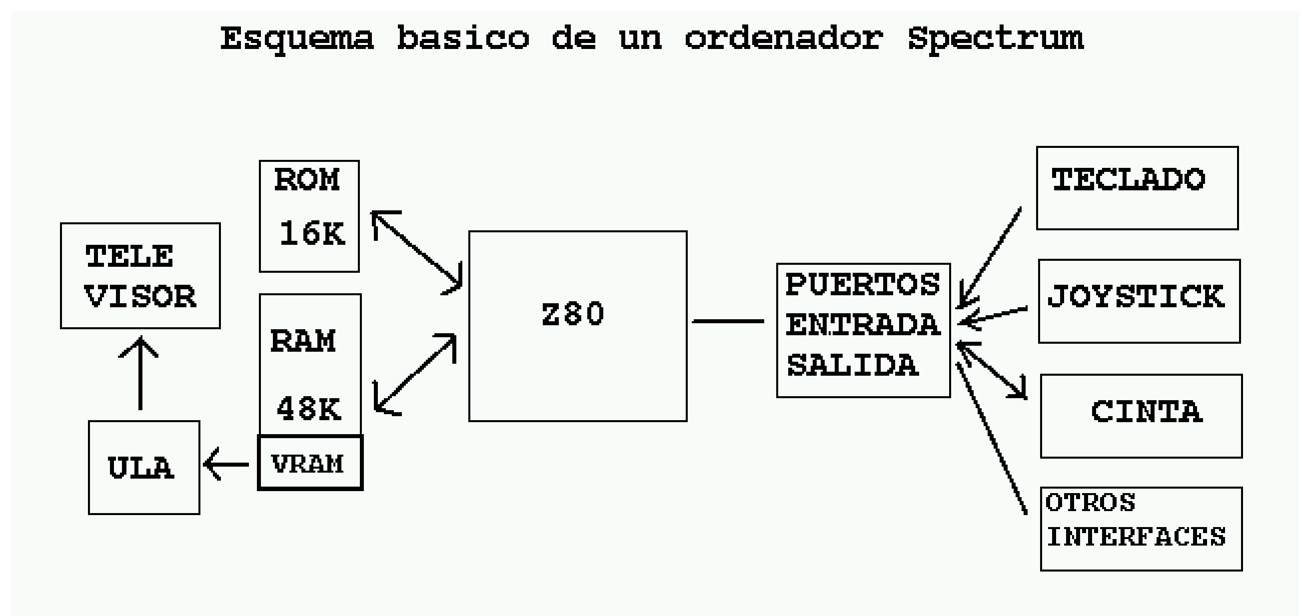


Figura 4. Esquema básico de un Spectrum real

La parte central es un micro Z80 el cual ve la ROM y la RAM de forma continuada como la totalidad de su memoria. Es decir, ve 64KB de memoria de los cuales los primeros 16K son el contenido de chip de ROM, y los siguientes 48K los del chip de RAM. Al encender el Spectrum éste se inicializa y muestra el BASIC porque comienza a ejecutar instrucciones desde la dirección de memoria 0, donde está el principio de la ROM de 16K, es decir, el intérprete BASIC.
Es por eso que cuando encendemos el ordenador se "ejecuta" la ROM. Al encender un Spectrum (que perdió en su apagado toda alimentación eléctrica) todos los registros del microprocesador Z80 valen 0, incluido uno que se llama PC (Program Counter o Contador de Programa), que es el que apunta a la siguiente instrucción que el Z80 debe leer y ejecutar. Un microprocesador funciona a grandes rasgos de la siguiente forma:

- Leer instrucción apuntada por el registro PC.
- Incrementar PC para apuntar a la siguiente instrucción.
- Ejecutar la instrucción recién leída.
- Repetir continuadamente los 3 pasos anteriores.

Así pues, al encender el ordenador, PC vale 0. Al estar la ROM mapeada en la posición de memoria 0 (mediante cableado hardware de los chips de memoria en la placa del Spectrum), lo que pasa al encender el ordenador es que ese contador de programa (PC) está apuntando al principio de la ROM, y es por eso que se ejecuta la ROM paso a paso, instrucción a instrucción, cada vez que lo encendemos. No hay misterio: para el Spectrum todos los chips de memoria de su interior son como si fuera un gran baúl de 64KB continuados, algo que se consigue mediante cableado de los diferentes chips a las patillas correctas del microprocesador. A grandes rasgos, las patillas de datos y de direcciones del microprocesador están conectadas a los diferentes chips de memoria de forma que cuando el micro lee

datos de la memoria, lo ve todo como si fuera un sólo chip de memoria de 64KB. Esto se consigue con un sencillo proceso de diseño (al hacer el esquema del ordenador antes de fabricarlo) conocido como "mapeado de memoria".

En el mapa de memoria del Spectrum, los primeros 16KB son la ROM (que está en un chip aparte, pero que como acabamos de ver es algo que el Spectrum no distingue, ya que la visualiza como una sección de memoria continua desde la posición 0 hasta la 16383 de su "baúl total" de 64KB) y luego viene la RAM, a partir de la posición 16384. Ahí es donde se almacenan los programas, los gráficos de la pantalla (en un trozo determinado de esa memoria), etc. En esta RAM es donde el intérprete de BASIC introduce los programas para su ejecución.
Estos programas pueden entrar desde los diferentes dispositivos de entrada/salida (gestionados por el Z80) como el teclado, la cinta o disco, etc.

Cabe hacer una mención especial a que una parte de la memoria RAM (desde el byte 16384 hasta el 23296) está conectada con la ULA, el chip "gráfico" del Spectrum, y encargado de convertir el contenido de esta "videoram" o VRAM a señales de vídeo para la televisión. Cuando los juegos dibujan gráficos, sprites o cualquier otra cosa en pantalla, en realidad están escribiendo bytes en estas posiciones de memoria, que la ULA muestra en la TV en el siguiente refresco de la pantalla.

Visto de una manera simple (pero real): al escribir un valor numérico (por ejemplo un 1) en alguna dirección de esta parte de la RAM, de forma inmediata aparece un punto en nuestro monitor, ya que la ULA está continuamente "escaneando" la videoram (de forma independiente del Z80) para reflejar en el monitor o televisión todos los valores numéricos que introduzcamos en ella. En los PCs ocurre igual: al escribir un valor numérico en una determinada posición de memoria, aparecen puntos en pantalla. Según en qué dirección escribamos aparecen en un lugar u otro de la pantalla. En algunos modos de vídeo (320x200, por ejemplo), escribir en la posición 0xA000 hace aparecer un punto de color en la posición (0,0) del monitor, hacerlo en 0xA001, lo hace aparecer en (1,0), y así un pixel tras otro.

Con esto obtenemos un software que emula un micro (el Z80) y que gracias a la ROM del mismo arranca y nos muestra y permite usar el Spectrum en sí mismo, tal y como se podría usar un Spectrum recién arrancado. El emulador lo que hace, en resumen, es leer los eventos de teclado y comunicarlos al micro emulado de la misma forma en que lo hacía el Spectrum real al pulsar una tecla. Al mismo tiempo, lee de la memoria del Spectrum el contenido de la pantalla, y lo muestra en nuestro monitor tal y como lo "leería" la Televisión. El microprocesador virtual, mientras tanto, se dedica simple y únicamente a ejecutar instrucciones, ya sea de la ROM o de un juego que carguemos o ejecutemos en él.

La ROM es muy importante, ya que indica cómo debe de comportarse el micro-procesador en todo momento, cómo debe atender las interrupciones recibidas por el teclado y los mandos, etc. Por ejemplo, un ordenador ZX Spectrum y un Amstrad CPC tienen el mismo microprocesador, un Z80, pero sin embargo son sustancialmente diferentes. ¿Por qué? Pues porque aparte de que los circuitos que acompañan al micro son diferentes, la ROM es totalmente diferente, de forma que cambia el Sistema Operativo, las direcciones de memoria donde se guardan los datos de pantalla, etc.

¿De dónde sale el software para los emuladores?

El mero hecho de poder utilizar una arquitectura determinada y su sistema operativo ya podría ser un gran aliciente, pero por si esto fuera poco, en la mayoría de los casos podemos además usar todo el software original del sistema físico en el emulador.

Ya sea un cartucho, una cinta o un disco, el objetivo es obtener una copia en formato digital (en un fichero) de sus datos, de forma que se puedan cargar en el emulador. Veamos los diferentes tipos de software a emular:

- ROMs del sistema: como ya hemos visto, las ROMs de las máquinas se obtienen

volcando el contenido de los chips de memoria de las mismas, donde están almacenados los programas en Código Máquina, mediante lectores de memorias o similares.



Figura 5. Una de esas pequeñas cucarachas negras (o varias, según el sistema) alberga la ROM del aparato.

- ROMs de cartuchos: los cartuchos de consola no suelen ser más que soportes de plástico de una forma determinada, que en su interior contienen chips de memoria (iguales que los chips de la ROM) conectados eléctricamente a los contactos metálicos o pines que sobresalen del plástico del cartucho. Si os fijáis, en realidad los juegos de consola no son más que ROMs del sistema, ya que realmente las consolas no suelen tener ROM (si las encendemos sin juego dentro, no hacen nada), y somos nosotros los que introducimos una ROM (que en realidad es el juego) al introducir el cartucho. Esa ROM, en lugar de ser un Sistema Operativo, es un juego. Recordemos que el microprocesador lo único que sabe hacer es ejecutar instrucciones en código máquina, sea un juego, o sea un intérprete de BASIC.



Figura 6. Cartucho de NES. En su interior sólo hay un pequeño circuito con una memoria ROM que contiene el juego grabado. En la captura podemos ver el circuito del juego Super Mario Bros 3 de Nintendo NES. Como puede verse, el juego está grabado en 2 chips de memoria y éstos vienen con la lógica necesaria para conectarse a la consola. Todos los chips están puenteados a las patillas del conector que se introduce en el cartucho. La carcasa gris de plástico tan sólo le da consistencia para su agarre e introducción en la consola, y para poner la portada del juego.

En algunos sistemas sí que tenemos una ROM pregrabada en la máquina capaz de realizar tareas cuando no introducimos ningún juego. De esta forma, las consolas clónicas de NES podían llevar cientos de juegos grabados en un chip ROM interno, de forma que cuando la encendemos sin introducir un cartucho, ese chip se activa y se ejecuta su contenido. Cuando insertamos un juego, por contra, el chip que realmente se convierte en la memoria de la máquina (y que por tanto se ejecuta) es el ubicado en el interior del cartucho.

Para volcar los juegos a ficheros de disco de forma que se puedan usar en los emuladores, basta con desmontar el cartucho, sacar el chip de memoria con el código del juego grabado, y al igual que en el caso de las ROMs del sistema, volcarlo a fichero con un lector de chips de memoria. Otra opción sería utilizar alguno de los lectores de cartuchos que se insertaban en la máquina y permitían grabar a disquete los contenidos de las ROMs. Estos aparatos fueron principalmente utilizados como "copiones".

En general, en el mundo de la emulación, se le llama ROM a todo fichero volcado a disco desde una memoria. Los ficheros de ROM tendrán diferentes formatos según la arquitectura de la máquina. Por ejemplo, fichero.smc será una ROM de cartucho de SuperNES, fichero.smd de Sega MegaDrive, fichero.nes un juego de NES, fichero.gb de Gameboy, etc. Así mismo, existen ROMs en el Spectrum, obtenidas de los cartuchos del Interface 2 (un periférico de Spectrum que permitía cargar juegos en cartucho, realizando exactamente la misma función que la inserción del cartucho en una consola).

Las máquinas recreativas son placas con microprocesadores y circuitos propios (al estilo de una consola) donde los juegos suelen estar grabados también en chips de ROM, o bien se introducen como si fueran cartuchos. El procedimiento para extraer las ROMs de las recreativas es similar al de los cartuchos en la mayoría de los casos.

- Cintas: Las cintas de Spectrum contienen software igual que los cartuchos de consola, sólo que en lugar de estar grabado el programa en un chip, lo está grabado de forma magnética en una cinta. La cinta contiene contiene audio grabado a una velocidad determinada y con un formato determinado para que el ordenador pueda leerlo (y grabar también software en cinta). El lector de cintas del ordenador lee este audio y lo interpreta, almacenando en memoria las instrucciones. Las cintas de Spectrum suelen ser ficheros con extensión .tap o .tzx, las cuales se han obtenido utilizando un programa de grabación de audio (con lo cual se guardan en disco en formato *.wav o *.voc). Un programa como Taper o Maketzx se encargará posteriormente de analizar este fichero wav/voc y convertirlo en un fichero de datos TAP o TZX con el programa en un formato perfecto y legible por los emuladores de Spectrum. Un fichero TZX contiene la cinta tal y como la tenemos grabada físicamente, pero en formato de unos y ceros en lugar de audio (por eso un TZX puede ocupar unos 100KB mientras que el WAV ocupa hasta 10MB, y son equivalentes). Lo más interesante es que a partir de un fichero TZX podemos obtener el WAV de la cinta (para regrabarla) o incluso es posible "reproducir" archivos TZX con programas como Playtzx, y cargar este juego en un Spectrum real con un cable macho-macho apropiado.

  Este formato de ficheros (TZX) es el más importante en el Spectrum. Esto es así porque al ser copias exactas de las cintas, y a partir de las cuales se puede regrabar una cinta original, al pasar a formato TZX todos los juegos disponibles, en realidad los estamos preservando y evitando la pérdida y deterioro de las cintas originales con el tiempo. Si una cinta, pasados los años, pierde sus propiedades magnéticas y el juego no carga, podemos regrabarla en la cinta original usando un grabador de cassette y el fichero TZX de la cinta obtenido de Internet.

  Hay que destacar que los ficheros TAP y TZX no son ROMS. Son "ficheros de

cinta" o "tape files". El término ROM, como ya hemos visto, se asocia a programas o juegos obtenidos desde hardware, y no es aplicable a estos ficheros.

- Discos: Al igual que las cintas, los discos de diferentes sistemas (incluidos los de 3 pulgadas) se pueden pasar a ficheros de disco con las unidades de disco apropiadas y programas que los interpreten y graben en los formatos adecuados. Por ejemplo, los ficheros .dsk, .tcl y .trd son imágenes de disco de Spectrum pasados a fichero y utilizables por los emuladores. Al igual que ocurre con las cintas, desde un DSK podemos recrear un disco de Spectrum en su formato original.

- Snapshots: son volcados de la memoria del Spectrum con los juegos ya cargados desde Cinta o Disco. Es decir, supongamos que en un Spectrum con 48KB de memoria (ZX Spectrum 48K) cargamos desde cinta un juego determinado. Este juego (que sin duda ocupará menos de 48KB) se almacena en la RAM del Spectrum listo para ser jugado (por ejemplo, en el menú del juego). Si en este momento grabamos el contenido de la RAM en un fichero, y al mismo tiempo almacenamos el estado completo de la CPU, tenemos una "copia" del estado de la máquina. Al cargar este fichero en otra máquina igual, o en un emulador, pondremos al sistema destino en el mismo estado exacto que estaba la máquina original. Es decir, delante del menú, o del juego, en el punto en que lo grabamos.

  Los Snapshots (que tampoco son ROMs) se suelen obtener cargando cintas en los emuladores y grabando el contenido de la memoria a disco con un formato especial según el tipo de Snapshot. Tenemos ficheros .z80 (que empezó a utilizar el emulador Z80 para MS-DOS), ficheros .sp y .sna (usados por emuladores antiguos), y ficheros .szx (el futuro posible formato estándar de snapshots), entre otros. Este formato de fichero no es el ideal para preservar los juegos, ya que no permiten recrear las cintas, tan sólo jugar de una forma rápida y directa a los juegos. En general podemos grabar y cargar snapshots en los emuladores de una forma muy rápida (como ficheros) y en cualquier momento, de forma que pueden ser usados como método para "grabar las partidas" y continuar posteriormente. En algunas Webs incluso se han acumulado Snapshots grabados justo antes de finalizar ciertos juegos, para que la gente pueda ver el final de un juego determinado sin tener que jugarlo (ver The Congratulations Archive en la sección de enlaces).

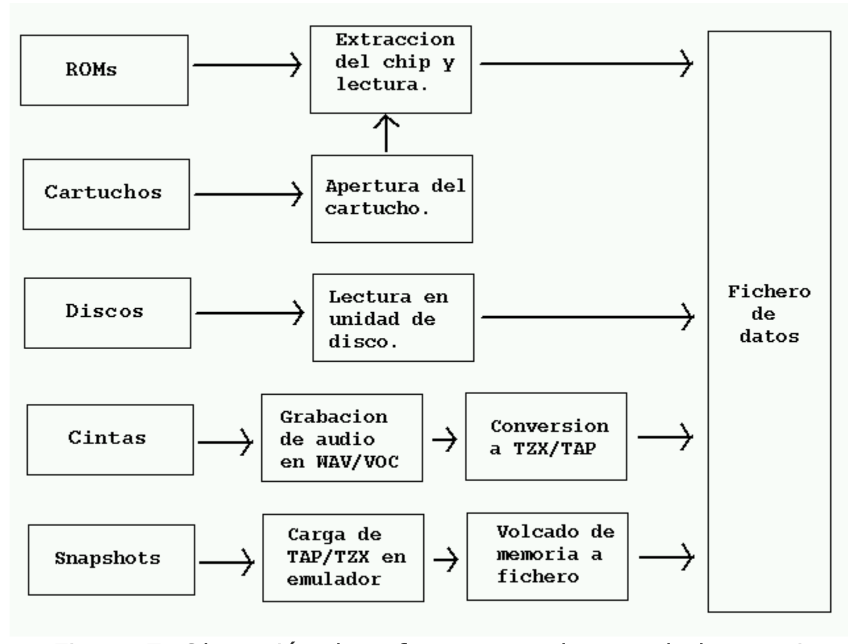


Figura 7. Obtención de software para los emuladores, visto de una manera simplificada

Aclaraciones

Existe una pequeña confusión entre 2 conceptos similares pero diferentes que suelen inducir a error: es la confusión entre "emulador" y "simulador". Hay 2 grandes definiciones sobre las diferentes entre ambos términos. La primera está basada en que el emulador es, como ya hemos visto, un programa más o menos complejo que emula la arquitectura de una máquina, sin apoyarse en nada más. Un simulador incorpora un emulador además de componentes hardware. Por ejemplo: un emulador de Spectrum emula todos los componentes del Spectrum vía software, mientras que un simulador de Spectrum podría ser un sistema con un micro Z80 conectable al puerto paralelo de un PC, de forma que el PC en vez de emular el Z80, le pidiera a éste la ejecución de las instrucciones, emulando el resto del sistema en el PC. Es decir, el simulador tiene una parte hardware, mientras que el emulador no.

En otros libros se realiza una descripción totalmente diferente, en la que el emulador es un dispositivo (software o hardware) que se encarga de implementar con la mayor exactitud posible todas las funciones del sistema emulado, mientras que el simulador es una caja negra que, independientemente de su contenido, se comportará igual que el sistema recreado. Es decir, ante un determinado conjunto de entradas proporcionará las mismas salidas que lo haría el sistema real.

Ambas definiciones se encuentran en diferentes libros de texto de Arquitectura de Computadores, y se pueden aplicar en diferentes ámbitos. La diferencia entre una u otra definición suele aparecer cuando se trata el tema de la Electrónica (momento en que se suele aplicar más la primera definición), frente a su uso en la Informática como desarrollo de software (donde se aplica más la segunda).

Otra pequeña confusión para los neófitos es aquella entre emulador y remake. Un remake es un juego programado desde cero de forma que se parezca a un juego antiguo o clásico de otro sistema. Por ejemplo, un remake de Manic Miner es un juego que tiene las mismas pantallas que el Manic Miner original de Spectrum, que simula los mismos personajes, enemigos y estilo de juego, pero que no está recreado mediante un emulador y el juego original, sino que está programado desde cero por uno o más programadores que realizan el juego con diferentes gráficos (aunque a veces pueden usarse los mismos capturados para darle un aspecto más fiel al original), mejores melodías y sonidos, etc. Los autores de remakes están en realidad realizando un juego como el original, pero hecho por ellos mismos. Con los remakes se consigue (además de satisfacer un reto personal y honrar al programador del juego original) acercar juegos clásicos a arquitecturas actuales, sin necesidad de emuladores.



Manic Miner original (izquierda) y uno de sus innumerables remakes (derecha)

Ventajas y desventajas de los emuladores

Existe un gran debate sobre la conveniencia o inconveniencia del uso y programación de emuladores. La gente se alinea directamente a favor o en contra de ellos según algunos de estos criterios:

A favor:

- Permiten utilizar sistemas que de otro modo no podríamos usar, por su no disponibilidad, su precio, etc.

- Permiten preservar todos los sistemas hardware mediante una emulación lógica por software (que es cada vez más perfecta) y que certifica que siempre podremos reutilizar los programas y juegos de los sistemas emulados, y además desde cualquier arquitectura actual y futura.

- Permiten encontrar más información sobre los sistemas emulados, y documentarlos de una manera efectiva para una posible futura reimplementación.

- Permiten probar sistemas que no conocías, y es posible que hagan que te aficiones a ellos, y trates de conseguir las máquinas propiamente dichas.

- Proporcionan facilidades al programador de estos sistemas para el desarrollo de aplicaciones y juegos para ellos, ya que éstos se pueden probar directamente en el emulador sin necesidad de pasarlos al sistema de prueba, acelerando la depuración de errores.

En contra:

- La utilización de emuladores hace que la gente se centre en la utilización de los mismos y la obtención de juegos y programas como ROMs con el único fin de acaparar juegos de todos los sistemas. Esto produce que la gente no se sienta realmente interesada en buscar, comprar y potenciar los sistemas físicos en sí mismos. Por ejemplo, ¿cuánta gente que use un emulador de SuperNintendo acaba comprando una SuperNintendo física y juegos para la misma? El disponer de los emuladores y las ROMs en CD en Internet hace que no se molesten en disfrutar del hardware en sí mismo.

- Los emuladores no son perfectos: no emulan al 100% todos los aspectos ni del microprocesador ni de los demás elementos asociados al sistema, por lo que la emulación (aunque sea casi perfecta) no es comparable a la sensación y *feeling* de la máquina real.

- El volcado de juegos que todavía se comercializan y la emulación de recreativas actuales y sistemas a la venta es dañino para el mercado del videojuego. Actualmente podemos jugar en emuladores Arcade a juegos de recreativa que todavía están en los salones. El disponer de estos juegos en casa hace que, al igual que ocurría en el primer punto enumerado, no hagamos uso de las máquinas pagando por usarlas.

La elección de una u otra postura es una cuestión personal, aunque el autor de este artículo puede decir sin tapujos que está a favor de la emulación, siempre que sea realizada de máquinas que actualmente no generen dinero en el mercado (es decir, recreativas antiguas, microordenadores y consolas de 8 y 16 bits). Estoy personalmente en contra de la emulación de recreativas modernas de última generación y sistemas actuales todavía a la venta (venta al por mayor, ya que el hecho de que todavía se vendan Atari 2600 en algún bazar no quiere decir que no se deban emular).

Resumiendo

Los emuladores son herramientas que nos van a permitir revivir tiempos pasados con sistemas que ya no están a nuestro alcance, o bien descubrir nuevas máquinas de las que nunca oímos hablar, o que nunca llegaron al lugar donde vivimos. Estos emuladores no son más que programas que emulan todos los aspectos hardware y software de la máquina con la mayor fidelidad posible. Al hacer esto, son capaces de entender lo que es la ROM de la máquina, y por tanto arrancar, y ejecutar las tareas de su "Sistema Operativo" básico (como por

ejemplo, la ROM del Spectrum con su BASIC) y por lo tanto entender también el software escrito para esta arquitectura, que es pasado a formato digital desde las cintas, discos o cartuchos donde estaban almacenados originalmente.

Espero que este pequeño artículo sirva como introducción a la emulación para aquellos que tenían dudas sobre ella.

LINKS

- Emulatronia (Juega con tu pasado):
  http://www.emulatronia.com
- Emudek:
  http://www.emudek.org
- Congratulations Archive:
  http://www.thespian.demon.co.uk/congrat/
- Cualquier utilidad de Spectrum como Taper, Maketzx o Playtzx:
  http://www.worldofspectrum.org/utilities.html#tzxtools
- Remake de Manic Miner para Windows:
  http://www.xmixdrix.com/manicminer/



MIGUEL G. PRADA

Segunda entrega de nuestro curso de juegos en BASIC, en la cual vamos a profundizar en qué son y cómo se usan las variables, las instrucciones que nos permiten tomar decisiones en nuestro programa, creación de subrutinas y el control del flujo de nuestro programa. Todo esto lo ilustraremos con algún pequeño ejemplo en forma de un sencillo juego con el que llevaremos a la práctica todos estos conceptos teóricos.

VARIABLES EN EL BASIC DEL SPECTRUM

Las variables son posiciones de memoria que se reservan para almacenar un determinado valor al que necesitamos acceder con facilidad desde nuestro programa y modificarlo a nuestro gusto. A estas posiciones de memoria se accede con el nombre que les damos al declarar las variables. Realmente poco nos importa en que posición de memoria las almacena nuestro ordenador, ya que siempre nos referiremos a ellas por su nombre.

- **DECLARACIÓN DE LAS VARIABLES**

Las variables tenemos que declararlas para poder usarlas, el ordenador no sabe ni el nombre ni el valor ni el tipo de la misma si no se lo decimos antes. Esto lo podemos hacer en cualquier lugar de nuestro programa siempre que lo hagamos antes de utilizarla. Para tener esto un poco más claro veamos el siguiente ejemplo:

- PROGRAMA 1 -

```
10 IF A=10 THEN LET A=A+1
```

No os preocupéis, de momento, por lo que hace la línea anterior.

Si ejecutamos la anterior línea con RUN, nos dará el siguiente error: `2 Variable not found, 10:1`. Esto quiere decir que no ha encontrado la variable que intentamos utilizar en la parte uno de la línea 10, o lo que es lo mismo, la variable 'A' justo después del IF. El error ha sido generado por no haber declarado antes de su uso la variable.

Añadamos al programa anterior la línea `5 LET A=10`, de manera que el programa quede de la siguiente forma:

- PROGRAMA 2 -

```
 5 LET A=10
10 IF A=10 THEN LET A=A+1
```

Lo volvemos a ejecutar con RUN y, en esta ocasión, nos devolverá el mensaje `0 OK, 10:2`, con lo cual nos dice que la ejecución del programa se ha llevado a cabo correctamente.

Si cambiásemos el número de la línea 5 por 15, nos volvería a generar un error de variable no definida, ya que declaramos la variable, pero después de hacer uso de ella.

- **TIPOS DE VARIABLES**

Las variables en el BASIC del Spectrum pueden ser de dos tipos: numéricas y alfanuméricas.

  - **VARIABLES NUMÉRICAS**

Al contrario de lo que pasa en otros lenguajes de programación, en los que tenemos diferentes tipos de variables para cada número que queremos almacenar, INT, FLOAT, DOUBLE, etc., en el BASIC del Spectrum no se distingue el tamaño de las variables numéricas. Al contrario, en ellas podemos almacenar cualquier tipo de número, ya sea entero o fraccionario, positivo o negativo.

El nombre por el que las vamos a designar debe comenzar obligatoriamente por una letra y puede estar seguida de más letras o números. Veamos los siguientes ejemplos de lo que son variables numéricas correctas e incorrectas:

Variables numéricas correctas

```
LET A=56
LET A1=-456
LET VIDAS=3
LET X1X=5.60
LET HOLA1N=-124.675
```

Todas estas declaraciones de variables son perfectamente válidas. Y las siguientes no podríamos declararlas:

Variables numéricas incorrectas

```
LET 1=45
LET 1A=-85
LET 4PLAYER=3
LET 1PUNTOS=5.5
```

Al intentar teclear estas instrucciones, el intérprete BASIC del Spectrum simplemente no nos dejaría introducirlas, generando un pitido y colocando el cursor en el punto erróneo.

A la hora de asignar los nombres a las variables numéricas, tenemos un par de excepciones que tener en cuenta: las variables que controlan los bucles y las que denominan a una matriz (a continuación explicamos qué es una matriz), que únicamente pueden estar nombradas con una sola letra:

`FOR F=1 TO 10: PRINT F: NEXT F` sería correcto y

`FOR F1= 1 TO 10: PRINT F1: NEXT F1` sería incorrecto.

Las variables que controlan los bucles no es necesario declararlas antes de utilizarlas, ya que en el mismo FOR se hace su declaración.

`DIM A(10,10)` válido y

`DIM A1(10,10)` inválido.

Las matrices se componen de una serie de valores almacenados en una cuadrícula a la que se accede por el nombre de la variable más su índice, cuya numeración comienza por uno. Es decir, si tenemos declarada una variable tal que `DIM A(10)`, podemos asignar a cada una de sus diez posiciones un valor diferente y utilizarlo de la misma forma que cualquier otra variable.

Por ejemplo, si queremos asignar a la quinta posición de nuestra matriz llamada 'A' el valor 345, nos bastaría con hacer: `LET A(5)=345` y lo podríamos utilizar, por ejemplo, de la siguiente forma: `IF A(5)>345 THEN CLS.`

Las matrices pueden tener más de una dimensión, podemos tener una *cuadrícula* de NxN elementos, sin más límite que la memoria del Spectrum. Imaginaos que definimos A(10,10), con lo cual tendremos una matriz de 10x10=100 elementos, cada uno almacenando un valor independiente, y a la que accedemos de la siguiente forma: `LET A(5,7)=10`. ¿Sencillo, verdad?

Imaginaos las aplicaciones que a una matriz se le puede dar en un juego. Valga como ejemplo el famoso juego de los barquitos, que los colocamos en una cuadrícula de NxN elementos, lo extrapolamos a una matriz y tendríamos el problema de cómo resolver su almacenamiento solucionado de una forma muy sencilla. Laberintos en un juego tipo Maziacs, las cartas en el Póker, etc.

- **VARIABLES ALFANUMÉRICAS**

El otro tipo de variables que tenemos disponibles son las alfanúmericas. En ellas podemos almacenar todo tipo de caracteres ASCII, es decir, números, letras y cualquier otro carácter o token del Spectrum.

Los nombres de las variables alfanuméricas solo pueden estar formados por una letra seguidos del símbolo '$', y no son válidas como índice de bucles, aunque sí para realizar comparaciones y diferentes operaciones de cadenas, como pueden ser concatenar unas con otras, elminar caracteres, etc. Este tipo de variables necesita que el valor que vayamos a asignarlas se indique entre comillas. Unos cuantos ejemplos:

Variables alfanuméricas correctas

```
LET A$="BUENOS DIAS"
LET A$="BUENOS"+"DIAS"
LET A$=CHR$ 65 (CHR$ meterá el
caracter ASCII correspondiente al
código 65, en este caso la letra 'A').
```

Todos los ejemplos anteriores son válidos y, al igual que anteriormente con los numéricos, los siguientes generarán un error:

Variables alfanuméricas incorrectas

```
LET A1$="BUENOS DIAS"
LET A$=BUENOS DIAS
```

También podemos crear matrices con contenido alfanumérico. Éstas tienen un peculiaridad muy importante. Siempre reservamos una longitud fija para almacenar la cadena. Mejor lo vemos con un ejemplo:

```
DIM A$(10,10)
```

Con esta instrucción le decimos a nuestro Spectrum que nos reserve memoria para una matriz de 10 elementos y con una longitud de 10 caracteres por elemento, es decir, podríamos almacenar diez 'frases' diferentes con una longitud de 10 caracteres y, si no rellenamos esos diez caracteres, el ordenador los rellena con espacios a la derecha de la cadena. Por ejemplo:

```
LET A$(2)="HOLA"
```

introduciría la palabra 'HOLA' (mejor dicho, sus códigos ASCII) en la posición 2 de la matriz, y rellenaría con seis espacios (carácter ASCII 32) los seis bytes restantes. Hay que tener en cuenta que cada carácter en ASCII tiene un byte de longitud.

- **CONSIDERACIONES FINALES SOBRE LAS VARIABLES**

Con las variables se pueden efectuar una amplia variedad de operaciones aritméticas, trigonométricas y lógicas: suma, resta, multiplicación, división, seno, coseno, etc., como pueden ser la siguientes:

`LET A=A+B` que suma el contenido de las variables A y B dejando el resultado en A.

`LET A=A+1` que suma a la variable A uno a su valor inicial.

`LET A=C*10` que pone en A el resultado de multiplicar por 10 el contenido de la variable C.

Y así con todas las operaciones disponibles.

Debemos tomar por costumbre nombrar a las variables con nombres que nos indiquen su utilidad, pero teniendo en cuenta que a mayor longitud del nombre, más espacio ocupan en memoria, y en el Spectrum la RAM es un bien escaso. Como ejemplo nos puede valer lo siguiente:

`VIDASJ1=3` con lo cual queremos indicar que el jugador 1 tiene tres vidas; esto ocupa, sólo por la definición del nombre, 7 bytes, sin contar el coste de almacenar el valor en memoria. En cambio si ponemos 'VJ1=3' lo reducimos en 4 bytes, esto puede parecer poco, pero si nos imaginamos un programa que utilice 30 variables, y sumamos los bytes gastados en la declaración más los que usemos cada vez que tengamos que realizar una operación con ella, podemos darnos cuenta de que se nos dispara el consumo de memoria.

TOMANDO DECISIONES

Cuando programemos nuestro juego, llegará un momento en que tengamos que hacer una cosa u otra dependiendo de determinadas condiciones que se den en el programa: si se nos terminan las vidas tiene que finalizar la partida, si matamos a un enemigo, que nos incrementen los puntos, etc. Esto se puede hacer en BASIC con los comandos que nos provee.

Para tomar una decisión hacen falta dos elementos que queramos comparar (o más en alguna ocasión) y un operador lógico que realice la comparación, es decir, podemos ver si entre dos variables una es mayor que la otra, si son iguales, si diferentes...

La principal instrucción que nos da el BASIC para efectuar comparaciones es `IF condición THEN`. Con esto conseguimos que, si el resultado de la comparación es cierto, se ejecuten los comandos que vienen a continuación del `THEN`. Lo veremos más claramente con un ejemplo:

```
10 IF VIDAS=0 THEN PRINT "ESTAS MUERTO":
STOP
```

Con esta instrucción comparamos si el contenido de la variable VIDAS es igual a 0, en caso afirmativo, se ejecutan los comandos situados después del THEN. En caso de que no sea cierta, simplemente se ignoran y se pasa la ejecución del programa a la siguiente línea del mismo.

Pero no sólo podemos realizar una comparación única por cada IF THEN que tengamos, podemos concatenar varias comparaciones:

```
10 IF VIDAS=0 OR TIEMPO=0 THEN PRINT
"ESTAS MUERTO": STOP
```

Aquí hacemos dos comparaciones y gracias al operador lógico OR, las instrucciones a continuación del comando THEN se ejecutan si cualquiera de los dos casos se cumple, que vidas o tiempo sea igual a 0.

Podemos tener la necesidad de que tengan que ser ciertas las dos, y no sólo una. Para que nuestra condición se cumpla:

```
10 IF VIDAS>0 AND TIEMPO >0 THEN LET
TIEMPO=TIEMPO-1
```

Fácil de comprender, si tenemos más vidas y tiempo que lo indicado, '0', ejecutamos las instrucciones colocadas después del THEN.

- **OPERADORES**

  Con estos elementos podremos comparar diferentes circunstacias entre variables. Y siempre devolverá la comparación si es cierto o falso.

  - Mayor que **">"** Cierto si el resultado de comparar el primer dato es mayor que el segundo.
  - Menor que **"<"** Cierto si el resultado de comparar el primer dato es menor que el segundo.
  - Igual que **"="** Cierto si los dos datos son exactamente iguales.
  - Diferente que **"<>"** Será cierto si los datos comparados son diferentes.
  - Mayor o igual que **">="** Cierto si el primer dato es mayor o igual que el segundo.
  - Menor o igual que **"<="** Cierto en caso de que el primer dato sea menor o igual que el segundo.

  Con este conjunto de operados lógicos podremos realizar las comparaciones necesarias en nuestro programa.

FLUJO DEL PROGRAMA

En el BASIC del Spectrum, las diferentes instrucciones de los programas las vamos introduciendo con un número delante, que nos sirve para poder controlar el flujo del programa como necesitemos durante la ejecución del mismo, entre otras cosas. La numeración de estas líneas la podemos indicar desde el número 1 al 9999, con los saltos intermedios que queramos, podemos numerarlas de 2 en 2, de 10 en 10, etc. Es aconsejable irlas numerando de manera que quede hueco entre una y otra para poder introducir una línea con instrucciones adicionales sobre la marcha, en caso de que sea necesario. Por ejemplo de 10 en 10.

En cualquier programa la ejecución de las instrucciones se va realizando de manera correlativa, en primer lugar la línea superior y consecutivamente las siguientes hasta llegar a la última. Esto para un programa sencillo puede servir, por ejemplo para uno que nos dé la equivalencia entre kilómetros y millas. Pero, en un juego, en el que hay porciones del programa que se ejecutarán varias veces, no es útil.

Tenemos dos instrucciones que nos van a permitir variar este flujo: `GO TO` y `GO SUB`.

- **GO TO**

  Con `GO TO` saltaremos a la línea indicada después de la misma instrucción, saltando las que pudiese haber tanto delante como detrás de la misma:

- PROGRAMA 3 -

```
 10 LET A=5
 20 IF A=5 THEN GO TO 100
 30 PRINT "NO ES CIERTO"
 40 STOP
100 PRINT "SI ES CIERTO"
```

  Al ejecutar este programa, como A=5, ya que así lo hemos asignado al principio del mismo, la condición analizada en la línea 20 se cumple y se ejecuta la instrucción GO TO, saltando directamente a la línea 100 sin pasar por la 30 ni la 40. Podemos variar ahora la línea 10 cambiando 5 por cualquier otro número. Veremos cómo la condición no se cumple y la ejecución del programa sigue por la línea 30 finalizando en la 40 por obra del STOP situado en ella, instrucción que sirve para detener la ejecución del programa drásticamente.

- **GO SUB y RETURN**

  La instrucción GO SUB es una evolución de GO TO ya que nos permite desviar en un momento dado la ejecución del programa, ejecutar un bloque de instrucciones y devolver el flujo a la línea siguiente donde se ejecutó el GO SUB con la instrucción RETURN. Este tipo de construcciones se denominan subrutinas.

  Lo vamos a ver con un ejemplo:

- PROGRAMA 4 -

```
10 LET A =5
 20 IF A=5 THEN GO SUB 100
 30 PRINT "DESPUES DEL GOSUB"
 40 STOP
100 PRINT "EL GOSUB HA FUNCIONADO"
110 RETURN
```

Es un programa similar al que usamos para ilustrar al comando GO TO. Si la condición de la línea 20 se cumple, que lo hará porque A vale 5 en un comienzo, salta a la línea 100, ejecuta lo que hay en ella y al llegar a la instrucción RETURN de la línea 110 , devuelve el flujo a la línea 30, justo después de la ejecución del GO SUB.

GO SUB nos va a ser muy útil a la hora de tener porciones de programa que se tienen que ejecutar varias veces, para no reescribirlos en varias ocasiones, simple-mente lo tenemos una vez y cada vez que nos haga falta lo llamamos con el correspondiente GO SUB.

Una precaución que tenemos que tener con el uso de RETURN, si lo ponemos en una línea, y el programa llega a ella sin haber ejecutado previamente una sentencia GO SUB, se detendrá la ejecución del mismo con el informe de error 7 RETURN without GO SUB, 0:1. Es decir, la ejecución de RETURN tiene que estar ligada forzosamente a una llamada anterior por medio de GO SUB.

Damos por supuesto que el lector ha comprendido que la utilización de GO TO y GO SUB no se limita al ámbito de una instrucción comparativa, sino que puede ser puesta en cualquier línea como un comando más para alterar el flujo del programa:

- PROGRAMA 5 -

```
 10 PRINT "HOLA MUNDO"
 20 GO TO 100
 30 REM A ESTA LINEA NO LLEGARA NUNCA
LA EJECUCION DEL PROGRAMA
 40 REM YA QUE EL GO TO HACE QUE LA
EJECUCION SALTE ESTA LINEA
100 PRINT "POR AQUI CONTINUA EL
PROGRAMA"
```

Vemos una instrucción nueva, REM, que sirve para insertar comentarios en nuestros programas, y que no se ejecuta en ningún caso. Nuestro programa se ejecuta escribiendo un saludo y luego salta directamente a la línea 100.

BUCLES

Podemos tener la necesidad de ejecutar una serie de instrucciones un número determinado de veces. Para eso tenemos los bucles, formados por los comandos FOR TO, NEXT y STEP.

El comando FOR necesita de una variable que irá incrementando o decrementando hasta que la condición de salida sea cierta y salga del mismo. Esta variable no es necesario que se declare previamente y su nombre constará de una única letra.

```
10 FOR F=1 TO 10
20 PRINT F
30 NEXT F
```

Este programa ejecutará 10 veces las instrucciones contenidas entre las instrucciones FOR y NEXT. En la línea 10 le decimos que vamos a crear un bucle llamado F que va a comprender un rango de 1 a (TO) 10, en la línea 20 ejecutamos lo que queremos, en este caso un PRINT y en la 30 cerramos el bucle.

La iteración del bucle no tiene por que ir de uno en uno, podemos hacer que la suma sea mayor con la inclusión del comando STEP. Cambiamos la línea 10 únicamente:

```
10 FOR F=1 TO 10 STEP 2
```

El bucle ahora se ejecutaría la mitad de veces, ya que le estamos indicando que incremente el valor de F de dos en dos, en lugar de uno como es lo habitual.

Para hacer que el bucle se ejecute hacia atrás, necesitaremos ponerle obligatoria-mente un STEP -1, ya que de otra manera, la iteración siempre suma. Cambiando la línea 10 del anterior programa por la siguiente podremos ver los resultados:

10 FOR F=10 TO 1 STEP -1

Podemos anidar un bucle dentro de otro sin ningún problema, siempre que nos preocupemos de cerrar el más interno antes de que se ejecute otra iteración del externo, Con un ejemplo lo veremos perfectamente:

- PROGRAMA 6 -

```
10 FOR F=1 TO 10
20 PRINT "TABLA DEL: ";F
30 FOR D=1 TO 10
40 PRINT F*D
50 NEXT D
60 NEXT F
```

Con este programa sacaremos por pantalla las tablas de multiplicar del 1 al 10, y podemos

apreciar como antes de cerrar el bucle externo, hay que hacer lo propio con el interno, en caso contrario el resultado obtenido no será el deseado. Se puede probar intercambiando las líneas 50 y 60.

NUESTRO PRIMER JUEGO

Para ilustrar todo lo que acabamos de leer, vamos a crear un jueguecillo simple de adivinación de números en el que vamos a utilizar las instrucciones vistas hasta aquí.

- ADIVINACIÓN -

```
 10  PRINT "DEBE ADIVINAR UN NUMERO
     ELEGIDO ALEATORIAMENTE ENTRE 1 Y 100
     EN LA CANTIDAD DE INTENTOS QUE USTED
     INTRODUZCA"
 20  INPUT "Introduzca el número de
     intentos (Max.10) ";A
 30  IF A<1 OR A >10 THEN GO TO 20
 40  LET VALOR=(RND*100)+1
 50  FOR F=1 TO A
 60  INPUT "INTRODUZCA NUMERO ";B
 70  IF B=VALOR THEN PRINT "ENHORABUENA,
     ACERTO EN ";F;" INTENTOS": STOP
 80  IF B>VALOR THEN PRINT "EL NUMERO
     INTRODUCIDO ES MAYOR"
 90  IF B<VALOR THEN PRINT "EL NUMERO
     INTRODUCIDO ES MENOR"
100  NEXT F
110  CLS: PRINT "NO HA SIDO CAPAZ DE
     ADIVINAR EL NUMERO EN ";A; "
     INTENTOS"
120  PRINT "EL NUMERO ERA ";VALOR
```

El programa es muy sencillo de entender, y solo haremos hincapié en unas cuantas instrucciones nuevas.

- INPUT, que como se habrá podido apreciar, sirve para introducir un valor por el teclado y asignarlo a una variable.
- PRINT, Sirve para presentar texto por pantalla.
- RND, que vale para generar números aleatorios, en este caso le decimos que busque un número entre 1 y 100 (como puede asignar 0 a este valor, le sumamos 1 para que nunca ocurra este caso).
- CLS, sirve para borrar la pantalla.
- STOP, con este comando paramos la ejecución del programa.

Algunas de estas instrucciones las comentaremos en profundidad en próximos capítulos del curso, de momento con saber qué hacen será suficiente.

CONCLUSIONES

Este mes hemos visto una serie de instrucciones que van a ser imprescindibles para realizar nuestros programas, así como dejamos sentados unos cuantos conceptos de programación sin los cuales no conseguiremos realizar un proyecto medianamente serio.

El juego de ejemplo de este mes puede no decir gran cosa, ya que carece de gráficos, efectos, movimiento, etc. pero sirve perfectamente para ver todo lo explicado en este artículo.

En la siguiente entrega de este curso, en la que abordaremos la presentación gráfica y la lectura de teclado, veremos moverse algo por la pantalla controlándolo a nuestro antojo.



MIGUEL G. PRADA

En este caluroso mes de Agosto en el que muchos estaremos disfrutando de unas merecidas vacaciones en la playa o en nuestro lugar de descanso favorito, vamos a analizar una web resucitada hace no mucho después de dos años de ausencia: El Mundo del Spectrum.

Todos tenemos en nuestra memoria la imagen dejada por EMS (**http://perso.wanadoo.es/spectrum/emds.html**) en su último número, el 36, publicado en web en Noviembre del año 2000. Un fabuloso monográfico dedicado a los juegos deportivos en nuestro Spectrum. Y como éste cada uno de los ejemplares que nos ofrecieron. Una publicación que nos tenía enganchados a muchos de nosotros, bien redactada, con un diseño gráfico espectacular, comentarios inteligentes y colaboradores de auténtico lujo.



La desaparecida EMS

Cuando Alejandro Ibáñez, creador de la la página, anunció su cierre después de cuatro años, 36 números y mucho tiempo sin actualizar, sentí algo similar a cuando aquel fatídico editorial de la MH 217 nos decía que se terminaba, un enorme vacío y la sensación de que una época finalizaba.

Y en este momento, después de un par de párrafos de nostalgia viva y de dar una pista de cómo era EMS, llegamos a la razón de ser de este artículo. Con el anuncio de la resurrección o nueva época unas cuantas preguntas surgen de repente: ¿cómo es ahora?, ¿qué puede aportar?, ¿es lo que era?.

EL DISEÑO

EMS 2.0 (**http://pagina.de/ems**) es un weblog basado en PHP, de estos que tan de moda están en la actualidad. Están pensados para ofrecer un dinamismo bastante alto a la hora de la publicación de noticias y de las posibles respuestas por parte de los lectores. A su vez, el mantenimiento que requieren es muy bajo.

El ambiente a Spectrum se respira sutilmente con el conocido "arcoiris" a cuatro colores típico del ordenador de Sir Clive enfatizando los diferentes recuadros de la web. Cada vez que accedemos nos muestra la foto de uno de los modelos de Spectrum existentes de forma aleatoria.

La página se ve correctamente en los navegadores habituales, tanto desde Linux como desde Windows (Internet Explorer, Netscape, Mozilla, Opera, Konkeror, etc.). Carga rápidamente, aunque tampoco es que tenga un tamaño exagerado para que tuviese que demorarse en exceso. La web está completamente en castellano, a expensas de las noticias que publique cada uno que, lógicamente, pueden estar en el idioma nativo del redactor.



EMS 2.0

LOS CONTENIDOS

Hablar de contenidos en este tipo de webs es complicado. No son sitios que se especialicen en alguna materia en concreto, como pueda ser recuperar programas en formato tzx, carátulas, manuales, etc. Se suelen limitar a la publicación de noticias u opiniones para abrir debate, y el hilo de contestaciones que se derivan de éstas.

Pero... y es un pero bastante lamentable por lo que nos toca a todos los que vivimos este mundillo: En esta web postea "SÓLO" el webmaster, las diferentes cuestiones no son practicamente leídas, no contesta "NADIE". Las votaciones que se ofrecen, con mayor o menor interés, son votadas por un número tan reducido de personas que en ningún lugar del mundo no serían válidas por falta de "quorum".

El hueco que quiere ocupar EMS en nuestro entorno está ampliamente cubierto por el grupo de news **es.comp.sistemas.sinclair** en castellano, **comp.sys.sinclair** en inglés y portales como **speccy.org** (como front-end a las webs que alberga y en el cual se suelen publicar las noticias más relevantes) o **microhobby.com**. Quizás la orientación no sea la idonea, pues no me cabe ningun atisbo de duda de los conocimientos de su creador, que dejó plasmados ampliamente en su anterior etapa, y de que hubiera podido hacer una web llena de contenidos con la experiencia que atesora.

Las noticias y cuestiones que presenta el webmaster están bien planteadas, ilustradas con capturas o fotos bien hechas, pero se queda en una especie de diario personal por la falta de activismo de todos nosotros.

EMS no tiene nada que ver con lo que era. De ser una publicación puntera y que creaba ansiedad por ver el próximo ejemplar, pasa a ser un weblog más. Y la diferencia es enorme.

LAS SECCIONES

Secciones, salvo la de "Links", no hay como tales. En la parte izquierda de la pantalla tenemos una serie de enlaces para visualizar las diferentes noticias ordenadas por categorias.

Y, como es normal en los weblogs, una sección de configuración en la que podremos adaptar el entorno de la web a nuestro gusto, siempre que estemos registrados como usuarios de la misma.

LO MEJOR

- Que su responsable sigue publicando noticias pese a que no son leídas, aunque con una cadencia cada vez más larga.

LO PEOR

- La falta de implicación de todos nosotros. La colaboración del público es forzosamente necesaria en este tipo de sitios.

EN RESUMEN

Después de dos años sin disfrutar de esa maravilla que era EMS (y añadir que, en opinión de quien esto escribe, deberían ponerse los 36 números anteriores a disposición de quien quiera disfrutar de ellos, bien sea en un CD o alojados en web), el anuncio de su reaparición hizo crecer en mi una ansiedad por ver que me encontraría en esta nueva andanza de EMS, y esta ansiedad se transformó en desilusión y extrañeza al ver EMS 2.0.

Una web que hoy por hoy no aporta nada nuevo a este mundillo, salvo trabajo a su creador que seguro la realizó con toda la ilusión puesta en ella y que no se ve recompensado en su esfuerzo.

PUNTUACIÓN: 4

Historia Ilustrada (y poco seria) del Spectrum

PEDRO J. DE CELIS

La situación económica de Sinclair era preocupante. La continuidad de la empresa estaba en entredicho. Se necesitaba una inyección de dinero de forma urgente, y no se descartaba la venta de la marca. Su rival en las islas británicas, Amstrad, lanzó una oferta de compra manteniendo el nombre de la marca. Sinclair salvaba los muebles a la vez que Amstrad hacía realidad uno de sus sueños: convertirse en la mayor empresa informática de Europa.







Las negociaciones sobre la compra de Sinclair fueron rápidas. En apenas dos meses se finiquitó la venta de Sinclair Research a Amstrad. La situación realmente desesperada de Sinclair y las ansias de Amstrad por convertirse en la empresa más importante del mercado ayudaron a que la transacción se resolviese con rapidez.

Bajo la mano de Amstrad, los ordenadores de Sinclair dieron otro giro. Amstrad decidió incorporar los elementos que tanto éxito habían cosechado en sus CPCs para los Spectrums: el Spectrum +2, con un reproductor de casette incorporado, y el Spectrum +3, con una unidad de disco 3" incorporada. A la vez que potenció al Spectrum, decidió liquidar otros proyectos interesantes, como el QL y el prototipo Loki de un SuperSpectrum. Ambos equipos podrían competir con su equipo estrella en el mercado doméstico: el CPC 6128.









¿Cómo se tomó la decisión de acabar con el QL? El ordenador era un sueño de Sinclair: un producto profesional a bajo coste, con unas características que lo hacían único para su época. Alan Sugar comprendió que hacerse la competencia a sí mismo no era algo muy inteligente. Había que sacrificar un producto, y el QL fue el elegido.

En los últimos años 80 el Spectrum había sido explotado hasta el máximo. Las técnicas de programación se habían perfeccionado. El mercado era enorme, por lo que las

producciones para el Spectrum eran muy numerosas. Las compañías crecían, y ofrecían cada vez mejores juegos, técnicamente más perfectos. Surgieron en esta época gran cantidad de compañías españolas que se añadían a las ya conocidas Opera, Dinamic, Topo, o Made in Spain. Nos estamos refiriendo a AD, Positive, Animagic, OMK, Iber, y GLL entre otros. Es el periodo conocido como "La Edad de Oro del Soft Español".







Fruto de este avance tecnológico, apareció publicado por Opera el que probablemente sea el mejor videojuego de la Historia del Soft Español. "La Abadía del Crimen" tenía todo lo necesario para convertirse en un gran éxito: excelentes gráficos, argumento interesante, y una puesta en escena espectacular. La única pega de este programa fue no haber sido distribuido en el extranjero, lo cual hace que este programa esté desaparecido de las listas de Mejores Programas que hay en la red.

El momento cumbre del Spectrum terminó con el cambio de década. La llegada de ordenadores más potentes, con mayores capacidades gráficas, y la bajada de precios de éstos hizo que las ventas de los micros de 8 bits decayeran. Los usuarios de Spectrum, hasta entonces tranquilos por la llegada de software en grandes cantidades, comenzaron a ver con preocupación la distribución de soft exclusivamente para las máquinas de 16 bits. Las noticias que llegaban de Europa comenzaban a ser desalentadoras. Una a una, cada compañía empezó a abandonar su línea de productos de 8 bits para centrarse exclusivamente en los Ataris, Amigas, y PC. Éste es el principio del fin.









A pesar de las mejores características técnicas de los ordenadores de 16 bits y de su bajada contínua de precios, estos sistemas no habrían tenido éxito si los usuarios de antiguos micros de 8 bits no se hubiesen pasado a los nuevos sistemas. Era normal en aquella época encontrarse con alguien mirando precios de los Amiga, o buscando juegos de Atari, que anteriormente habían tenido un Spectrum en sus manos.

La situación se vino abajo a principios de los 90. Los desarrollos de las compañías para el Spectrum iban desapareciendo. Amstrad dejó de fabricar el Spectrum +3 en 1991. En 1992 sólo Ocean y alguna compañía budget que no se distribuía fuera del Reino Unido hacían soft para el Speccy. Las revistas fueron desapareciendo una a una. Se ha considerado el final de la época del Spectrum el año 1992. Por aquel entonces, el Spectrum era un ordenador comercialmente muerto.



En enero de 1992 salió el que sería el último número de la revista más importante del panorama del Spectrum en España. La Microhobby desapareció, tras 7 años de puntualidad en los kioscos, para dar paso a otras publicaciones de la misma editorial encaminadas al mercado de las consolas. Con su desaparición, los usuarios de Spectrum en España se quedan huérfanos y desamparados. En ese año ya no existe nada relacionado con el Spectrum que pudiera comprarse en una tienda.

¿Qué le ocurrió al Spectrum tras su muerte comercial?
¿Comprará Alan Sugar el Manchester United?
¿Se comprará Sir Clive Sinclair un Amiga?
¿Se reeditará la MicroHobby en formato papel higiénico?

En la próxima entrega, la 3ª (y última) parte de la Historia Ilustrada (y poco seria) del Spectrum.